Incorporating Guard Instructions into Existing Architectures

by

Selim Bilgin

A thesis submitted in partial fulfillment
of the requirements for the Degree of

Master of Science
(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN - MADISON

1997

This thesis has been approved by

Gurindar S. Sohi

Professor of Computer Sciences and Electrical and Computer Engineering
University of Wisconsin - Madison

____

Date

Incorporating Guard Instructions into Existing Architectures

Selim Bilgin
UNIVERSITY OF WISCONSIN - MADISON, 1996
Under the supervision of Gurindar S. Sohi

Guarded Execution, as a concept, has been investigated in several studies emphasizing its potential to reduce the unpredictability of the control flow caused by branches. It has also been stated that having *Guarded Execution* exploits Instruction Level Parallelism (ILP). However, it is not straightforward to integrate *Guarded Execution* in an existing architecture.

This thesis studies the integration of *Guarded Execution* implemented by *Guard Instructions* in an existing in-order and an out-of-order pipeline, using a DLX type instruction set. In both cases, first the original pipelines are designed using Verilog HDL, and then the additional hardware needed is added, finally both these versions are synthesized using Synopsys' synthesis tools. The thesis starts with a brief overview of *Guarded Execution* and *Guard Instructions* followed by the hardware analysis.

The results of this thesis show that the hardware overhead of having the three types of *Guard Instructions*, namely *Guard_True*, *Guard_False* and *Guard_Both* is in a justifiable range of 15% in terms of area for both cases; and 1.21% in terms of cycle time for the out-of-order case. The results also show that no cycle time overhead has been encountered in the in-order case.

Acknowledgments

I would like to thank my advisor Guri Sohi who has taught me a lot since the good old 552 days. I also would like to thank Jim Smith for his support and understanding.

I also want to thank Dionisios Pnevmatikatos for his wonderful Ph.D. research which led me the way through this work.

I also would like to thank all the 3652ers, especially Eric Rotenberg, Quinn Jacobson and Timothy Heil who were all there for me whenever I had a problem, anykind!

Finally, how can I help but thank my family. Dad, mom; thank you very much for making anything I could and couldn't imagine come true; thank you very much for your love and guidance; and most important of all thank you very much for your understanding. This one's for you...

SELIM BILGIN

UNIVERSITY OF WISCONSIN - MADISON
May 1997

Contents

Abstract **i**

Acknowledgments **ii**

List of Tables **v**

List of Figures **vi**

Chapter 1 Introduction **1**

Chapter 2 GUARD Instructions **3**

2.1 Guarded Execution and Guarded Instructions..3

2.2 Cost Issues in Guarded Execution..5

2.3 Summary...7

Chapter 3 The Instruction Set and Modifications For Guarding **8**

3.1 The Instruction Set..8

3.2 Format of GUARD Instructions..10

3.3 The Operation of GUARD Instructions..11

3.4 Summary...15

Chapter 4 Integrating GUARD Instructions into an In-Order Pipelined Architecture **16**

4.1 Properties of the Modeled Pipeline...16

4.2 The Additions and Modifications to the Hardware for GUARD Support...20

4.2.1 Decoder Logic for the Mask Field of GUARDBOTH Instructions...20

4.2.2 Shift Register Design for Support of Guarding.......................................21

4.2.3 Area, Gate Count and Timing Overheads..30

4.3 Conclusions...36

Chapter 5 Integrating GUARD Instructions into an Out-of-Order Pipelined Architecture **39**

5.1 Properties of the Modeled Pipeline...39

5.2 The Additions and Modifications to the Hardware
for GUARD Support....43
5.2.1 Value Field Generator Logic....44
5.2.2 Control Logic for GUARD Instructions....45
5.2.3 The Shift Register....47
5.2.4 Area, Gate Count and Timing Overheads....47
5.3 Conclusions....55
Chapter 6 Conclusions **57**
6.1 Future Work....58
References **59**
Appendix 1
LCA500K Internal Macrocells Used....60
Appendix 2
Synthesis Results of the In-Order Pipelined Architecture....64
Appendix 3
Synthesis Results of the Out-Of-Order Pipelined Architecture....102

List of Tables

3.1 The instructions, opcodes and function fields used.......................................9

3.2 Encoding scheme used for GUARDBOTH instructions.......................................13

3.3 The opcodes for GUARD type instructions.......................................14

4.1 The SRAM Portions of the Modeled Pipeline.......................................18

4.2 Synthesis results for Each Stage of the Modeled Pipeline.......................................19

4.3 Modified pipeline synthesis results for minimum area.......................................32

4.4 Gate types, counts and the area occupied by the Mask Decoder.......................................32

4.5 Gate types, counts and the area occupied by the added hardware to EX stage.......................................33

4.6 Gate types, counts and the area occupied by the Shift Register.......................................35

4.7 Percent overhead of GUARDed Execution for each stage.......................................37

5.1 The SRAM Portions of the Modeled Pipeline.......................................42

5.2 Synthesis Results for Each Stage of the Modeled Pipeline.......................................43

5.3 Modified pipeline synthesis results for minimum area.......................................50

5.4 Gate types, counts and the area occupied by the Value Field Generator.......................................50

5.5 Gate types, counts and the area occupied by the Control Logic.......................................51

5.6 Gate types, counts and the area occupied by the Shift Register.......................................53

5.7 Percent overhead of GUARDed Execution for each stage.......................................55

A.1 Macrocell Abbreviations.......................................60

List of Figures

2.1 C code example....4

2.2 Compiled code for the C code in Figure 2.1....4

3.1 The formats of the instructions in DLX Instruction Set....9

3.2 GUARD Instruction Format....10

3.3 The usage of GUARD instructions....14

4.1 Basic structure of the pipeline modeled....17

4.2 The logic and the critical path of the Mask Field Decoder for GUARDBOTH....22

4.3 The logic diagram and the critical path of the hardware added to the EX stage....26

4.4 Same as Figure 4.3., without the 32-bit OR gate....27

4.5 The circuit diagram of the "One's Position Detector Logic"....29

4.6 The interface to the Shift Register....30

4.7 The circuit diagram of the Shift Register....31

5.1 Data flow diagram of the out-of-order pipelined design....40

5.2 The logic and the critical path of the Value Field Generator Logic....46

5.3 The logic and the critical path of the Control Logic for GUARD Instructions....48

5.4 The logic and the critical path of the Shift Register....49

Chapter 1

Introduction

There has been, is and will always be a demand for faster processors. This need for faster processors has introduced processors which have the ability to exploit the Instruction Level Parallelism in a given code. The designers also are working on many ways to reduce the clock period of the processors; and usually achieve this task by dividing the work that is originally to be done in one clock cycle and completing it in more than one cycle; this technique is widely known as *pipelining*. It is obvious that in order to get more out of this scheme, the machine should have the instructions flow through its pipeline in a smooth fashion; in other words, the more stalls there are, the more performance degradation the designers are facing.

There are more than one reasons for stalling the pipeline, one of the more important ones being the *control dependences*. Changes in the control flow of a program can introduce stalls during runtime since the processors cannot determine which instruction is to be fetched next. Especially when the number of stages in a pipeline increases, the time needed to resolve this dependency in terms of clock cycles increases and this results in a poor performance. This effect has been intensively studied and several static and dynamic branch prediction schemes have been proposed [7].

In many previous studies, "*Guarded Execution*" is shown to be an alternative to deal with control dependencies [1][2][9][10]. *Guarded Execution* decides the instructions to be executed by testing a *guard operand*. Although it gives a flavor of control flow change, the flow is still sequential and the instructions are fetched sequentially; and moreover the stalls due to the control dependencies are eliminated. Since there is no prediction involved for the instructions that are guarded, there can be no mispredictions for these instructions, hence no correction operations are needed.

Since nothing can help us achieve the ideal easily, there are some disadvantages of *Guarded Execution*. The first one is about performance; some instructions that are guarded might have already been fetched and might be flowing through the pipeline and depending on the guard operand they might have to be converted to NOPs. This results in wasting some fetch bandwidth, and in having bubbles floating through the pipeline. The second disadvantage is due to hardware; since the region that can be guarded is limited to some extent, the branch instructions which change the control flow of the program are still needed, hence some sort of branch prediction scheme is also still needed. Moreover, there is some additional hardware required for the implementation of *Guarded Execution*. This hardware overhead for *Guarded Execution* has not been investigated in studies prior to this thesis.

This thesis presents the study of the hardware cost issues encountered in the integration of *GUARD Instructions* into existing in-order and out-of-order pipelines running DLX-like code [5]. Chapter 2 gives a brief description of *Guarded Execution* and *Instruction Guarding* concepts. Chapter 3 defines the base instruction set and the *GUARD* type instructions introduced to it. Chapter 4 investigates the integration of the *GUARD* instructions into an In-Order Pipelined architecture. Chapter 5 investigates the integration of the *GUARD* instructions into an Out-Of-Order Pipelined architecture. Chapter 6 states the conclusions and concludes the work with possible work to be done in the future.

Chapter 2

GUARD Instructions

In this chapter, a brief definition of *Guarded Execution* is given with a simple example after which the cost issues regarding the use of *Guarded Execution* are presented.

2.1 Guarded Execution and Guarded Instructions

Guarded Execution means selectively executing instructions depending on the value of a *condition register*. An instruction which might be turned into a NOP depending on the value of the *condition register* is called a *Guarded Instruction.* If the condition meets the requirement, then the instruction can be executed; if it does not, then the instruction is dynamically turned into a NOP. Since *Guarded Execution* can turn instructions into NOPs, there is no more need for conditional branches which can branch ahead upto some number of instructions in the sequential code. This decreases the number of branch instructions in the code substantially resulting in a decrease in the number of control dependences. Hence the Instruction Level Parallelism in the code is increased.

Several commercially successful architectures like CRAY-1, DEC Alpha and Intel P6 have supported some form of *Guarded Execution.*(DEC Alpha and Intel P6 have limited their guarding concepts to '*conditional move*' type instructions).

A *Guarded Instruction* looks like:

g_and   cond ? rd, rs1, rs2

meaning that the destination register (*rd*) will be modified with the result of the bitwise AND operation (*rs1 && rs2*) if and only if the condition specified in the *cond* field evaluates to be true; as opposed to the non-guarded version of the same AND instruction which modifies the

```
for (i = 0; i < N; i++) {
    if (A[i] == 0) {
        B[i] = 0;
    }
    else if (A[i] < 0) {
        B[i] = -1;
    }
    else {
        B[i] = 1;
    }
}
```

Figure 2.1 C code example

specified destination register regardless of a condition.

We can go through a simple example which shows how the number of control dependences is reduced by *Guard* type instructions. If we consider the loop in Figure 2.1,

```
L0:  LW      r3, 0(r1)
     BEQ     r3, r0, L1
     SLTI    r5, r3, #0
     BNE     r5, r0, L2
     SW      r9, 0(r2)
     J       L3
L1:  SW      r0, 0(r2)
     J       L3
L2:  SW      r8, 0(r2)
L3:  ADDI    r1, r1, #1
     ADDI    r2, r2, #1
     BNE     r1, r7, L0
```

(a)

```
L0:  LW      r3, 0(r1)
     G_SW    ~r3? r0, 0(r2)
     G_J     ~r3? L1
     SLTI    r5, r3, #0
     G_SW    ~r5? r9, 0(r2)
     G_SW    r5? r8, 0(r2)
L1:  ADDI    r1, r1, #1
     ADDI    r2, r2, #1
     BNE     r1, r7, L0
```

(b)

Figure 2.2 Compiled code for the C code in Figure 2.1

which realizes the SGN (sign) function on an array of real numbers and writes the result into the result array. A simple compiler can generate the MIPS like code given in Figure 2.2.a; having the contents of the registers are as follows: r1 contains A[0], r2 contains B[0], r7 contains N, r8 contains -1, r9 contains 1 and r0 is hardwired to 0 as in the MIPS architecture family [3][4][13].

And in Figure 2.2 b, the same loop compiled with the *Guarded Instructions* is shown. The very first thing to be noticed is the reduction in the number of conditional branches from 3 to 1. The next thing that can be seen is the reduction of the total instruction count, from 12 to 9. If a single cycle load latency, a single cycle branch delay (in the machines with no static or dynamic branch prediction) and equal instruction and data cache hit percentages are assumed, then the code in Figure 2.2.a will execute in 13 cycles when the longest path is traversed and in 10 cycles when the shortest path is traversed. On the other hand, the code in Figure 2.2.b will execute in 10 cycles; resulting in an average 1.2 speedup if all the three paths have equal probability to be traversed. Even if the machine has some sort of a dynamic branch prediction scheme, since the branch predictors can not be ideal there will be some mispredictions, probability of which is relatively higher for the first two branch instructions in the code in Figure 2.2.a. These two branch instructions do not exist in the code in Figure 2.2.b, therefore there will be no mispredictions due to these instructions in the *Guarded* version, which results in a less misprediction percentage in the overall code.

2.2 Cost Issues in Guarded Execution

There are several cost issues regarding the implementation of *Guarding.* As mentioned above, there are two different instructions for the same operation, namely the *guarded* and the *unguarded* versions. This requires support from the *Instruction Set* since the opcodes of these instructions should be different.

Another cost issue is the increase in the total number of instructions that are actually fetched and injected into the pipeline. All the instructions in the code are fetched since the machine has no way to identify the instructions that are actually going to be executed and the ones that are going to be turned into NOPs in advance. This property of *Guarded Execution* can easily be seen in Figure 2.2. The *Guarded* version of the code has only one path for control flow while the original version can skip some instructions from label(s) L0 and/or L1 and/or L2. Some clever scheduling algorithms like scheduling the instructions converted to NOPs in parallel with instructions that actually do useful work has been studied in [1], and it is shown that using this kind of scheduling algorithms reduces the execution time.

One other difficulty in implementing *Guarded Execution* is the need to evaluate the *Guard Condition* by reading the *Guard Condition Register*. Since this register is needed at the same time that the source registers are needed, the number of read ports of the *Register File* should be increased. It is widely known that increasing the number of ports of an SRAM block increases the access time to the block; hence slows down the cycle time.

The last issue is that the registers that are used in branch instructions for comparison are needed only for the lifetime of the branch instructions and can be modified as soon as the branch instruction retires, but this is not the case for the *Guarded* version of the same code. This property of *Guarded Execution* makes the compiler's job a little harder. Another solution to this problem would be adding a separate register file to the architecture just for the *Guard Condition Registers*. However this solution would require a major design change, hence will not be justifiable when trying to add and modify the minimum hardware.

2.3 Summary

It has been shown [1] that *Guarded Execution* is a powerful and promising concept. The next chapters show that the amount of hardware required to support *Guarded Execution* is also reasonable, which makes it even more attractive to include in an instruction set.

The following chapters present the integration of *Guard Instructions* [1] into existing architectures starting with the brief description of the instruction set and *GUARD Instructions* that is used in this work.

Chapter 3

The Instruction Set and Modifications For Guarding

The previous chapter briefly introduced the issues encountered in integrating *Guarding* into an existing architecture. It has been shown in [1] that, in order to support *Guarding* the existing instruction set should be modified by introducing some new instructions, and the conclusion of the study suggests introducing three new instructions, namely *GUARDTRUE, GUARD-FALSE* and *GUARDBOTH* instructions. In this work, these instructions are introduced to the existing 32-bit reduced DLX-instruction set.

Since the change involves adding just three new instructions, some of the reserved opcodes of the DLX instruction set can be used for this purpose. And since the opcodes of the other instructions are left unchanged, the cost issue related to the opcode space which was mentioned in Chapter 2 becomes irrelevant.

3.1 The Instruction Set

The instructions that are used in this work have the formats that are shown in Figure 3.1. The instructions that are modeled in this work excluding the three *GUARD* instructions are shown in Table 3.1.

Figure 3.1 The formats of the instructions in DLX Instruction Set

I-TYPE INSTRUCTION

6	5	5	16
OPCODE	rs1	rd	IMMEDIATE

R-TYPE INSTRUCTION

6	5	5	5	11
OPCODE	rs1	rs2	rd	FUNCTION

J-TYPE INSTRUCTION

6	26
OPCODE	OFFSET ADDED TO PC

Table 3.1: The instructions, opcodes and function fields used

Instruction	Type	OPCODE[5:0]	FUNCTION[10:0]
ADD	R	0x00	0x020
ADDI	I	0x08	N/A
ADDIU	I	0x09	N/A
ADDU	R	0x00	0x021
AND	R	0x00	0x024
ANDI	I	0x0c	N/A
BEZ	I	0x04	N/A
BNZ	I	0x05	N/A
J	J	0x02	N/A
JAL	J	0x03	N/A
JALR	R	0x00	0x009
JR	R	0x00	0x008

Table 3.1: The instructions, opcodes and function fields used

Instruction	Type	OPCODE[5:0]	FUNCTION[10:0]
LHI	I	0x0f	N/A
LW	I	0x23	N/A
OR	R	0x00	0x025
ORI	I	0x0d	N/A
SLL	R	0x00	0x004
SLT	R	0x00	0x02a
SLTI	I	0x0a	N/A
SRA	R	0x00	0x007
SRL	R	0x00	0x006
SUB	R	0x00	0x002
SUBU	R	0x00	0x023
SW	I	0x2b	N/A
XOR	R	0x00	0x026
XORI	I	0x0e	N/A

3.2 Format of GUARD Instructions

All the *GUARD* instructions that are used in this work have two operands; the first being the *Guard Condition Register,* and the second being the mask field of the *GUARD* instruction. The format of the *GUARD* instruction is shown in Figure 3.2.

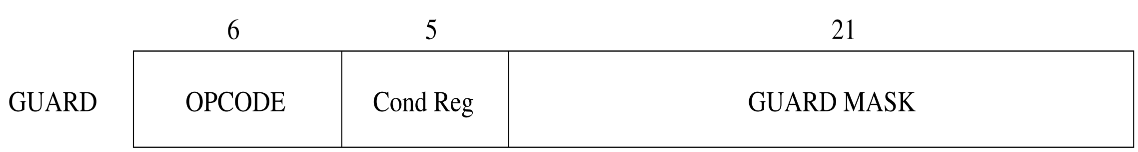

Figure 3.2 GUARD Instruction Format

3.3 The Operation of GUARD Instructions

The operation of the *GUARD* instructions is as follows: first the *Guard Condition Register* is fetched, then evaluated, and if the evaluation turns out to be false then the instructions that are listed in the *Guard Mask* are turned into NOPs. However, the way that the *Guard Mask* works for all the three *GUARD* instructions is not the same. Throughout this work, a *Condition* is said to be false if the content of the *Condition Register* is zero; and true otherwise.

A *GUARDTRUE* instruction looks like:

GUARDTRUE   cond, i 21, i 20, ... , i 2, i 1

where (i 1, i 2, ...) are the labels of the instructions that are preceded by the *GUARDTRUE* instruction (i.e. ‘i 1’ is the sequentially very next instruction that follows the *GUARDTRUE* instruction).

The way that the *Guard Mask* works for the *GUARDTRUE* instruction is such that if a mask bit is reset (‘0’) then the instruction corresponding to this bit is not *Guarded* and hence will not be affected by the outcome of the *GUARDTRUE* instruction; on the other hand if the mask bit is set (‘1’) then the instruction is *Guarded.*

The instructions that have their mask bits set are to be converted to NOPs when the *Guard Condition Register* content is false.

A *GUARDFALSE* instruction works in a similar way as the *GUARDTRUE* does. The only difference is that the *Guarded* instructions will be converted to NOPs if the *Guard Condition Register* content is true.

The third *GUARD* type instruction is the *GUARDBOTH* instruction. The difference of this instruction from the other two is its ability to guard any combination of the instructions that are listed in its *Guard Mask* under the *Condition Register* being true or false.

The result of the study in [1] suggests that there is more involved in the hardware implementation of the *GUARDBOTH* instructions. The *Mask Field* of the *GUARDBOTH* instruction must specify three states for each instruction; namely ‘not guarded’, ‘guarded on true’ and ‘guarded on false’. So, at least 2 bits are needed for representing the state of an instruction. This scheme will limit the number of instructions that can be guarded by the *GUARDBOTH* instruction to 10 since the mask field of a *GUARD* type instruction consists of 21 bits. But this scheme wastes the fourth unused state for each instruction. Instead, one can use some kind of an encoding scheme to take advantage of this fourth unused state.

Since for three instructions the number of possible guard states is 3^3=27 which can be represented in binary form by 5 bits instead of 6 which is needed by the previous ‘2 bits per instruction’ scheme needs. Using this scheme, one can use

$$\left\lfloor \frac{21}{5} \right\rfloor = 4$$

encoded blocks representing 3*4=12 instructions instead of 10. This scheme leaves the 21^{st} bit of the mask field unused; and the overhead is just a relatively simple decoder circuit just before the actual *Guard Mask* is needed.

The optimized encoding scheme which requires the minimum amount of logic for its decoder and is used in this work for the mask field of the *GUARDBOTH* instruction is shown in Table 3.2. In the table, X corresponds to ‘not guarded’, 1 corresponds to ‘guarded on true’ and 0 corresponds to ‘guarded on false’.

Table 3.2: Encoding scheme used for GUARDBOTH instructions

Mask[4:0]	Inst 3	Inst 2	Inst 1
0x00	X	X	X
0x04	1	X	X
0x0c	0	X	X
0x02	X	1	X
0x0a	X	0	X
0x01	X	X	1
0x09	X	X	0
0x06	1	1	X
0x0e	0	0	X
0x05	1	X	1
0x0d	0	X	0
0x03	X	1	1
0x0b	X	0	0
0x16	1	0	X
0x1e	0	1	X
0x15	1	X	0
0x1d	0	X	1
0x19	X	1	0
0x1b	X	0	1
0x07	1	1	1
0x0f	0	0	0
0x17	1	1	0
0x1f	0	0	1
0x11	1	0	1
0x13	0	1	0
0x10	0	1	1
0x12	1	0	0

The opcodes that are used for these three *GUARD* type instructions are given in Table 3.3.

Table 3.3: The opcodes for GUARD type instructions

Instruction	Opcode[5:0]
GUARDTRUE	0x19
GUARDFALSE	0x1a
GUARDBOTH	0x18

In Figure 3.3.a, the recompilation of the example in Figure 2.1 is shown allowing *GUARDTRUE* and *GUARDFALSE* instructions only, and in Figure 3.3.b, the *GUARDBOTH* instruction is also allowed.

```
L0:  LW    r3, 0(r1)
     GUARDFALSE   r3, 00003
     SW    r0, 0(r2)
     J     L1
     SLTI  r5, r3, #0
     GUARDFALSE  r5, 00002
     GUARDTRUE   r5, 00002
     SW    r9, 0(r2)
     SW    r8, 0(r2)
L1:  ADDI  r1, r1, #1
     ADDI  r2, r2, #1
     BNE   r1, r7, L0
```

(a)

```
L0:  LW    r3, 0(r1)
     GUARDFALSE r3, 00003
     SW    r0, 0(r2)
     J     L1
     SLTI  r5, r3, #0
     GUARDBOTH r5, 00019
     SW    r9, 0(r2)
     SW    r8, 0(r2)
L1:  ADDI  r1, r1 #1
     ADDI  r2, r2, #1
     BNE   r1, r7, L0
```

(b)

Figure 3.3 The usage of GUARD instructions

One can easily observe that even in a simple loop like this, having *GUARDBOTH* instruction available reduces the total instruction count by one per iteration, and hence the total execution time.

3.4 Summary

As long as there are at least three reserved opcodes in an existing instruction set, the three *GUARD* type instructions can be introduced to that instruction set.

In this chapter, the reduced DLX instruction set that is used in this work with the *GUARD* type instructions and their operations were described. In the Chapter 4, the actual integration of the *GUARD* type instructions into an existing in-order pipelined machine is investigated.

Chapter 4

Integrating GUARD Instructions into an In-Order Pipelined Architecture

In this chapter, the costs of integrating the three *GUARD* type instructions into an existing in-order pipelined architecture is investigated. The modeled machine is a simple in-order pipeline which can execute the instructions given in Table 3.1. The pipeline is structurally implemented using Verilog HDL except its memory components such as the register file, the data cache and the instruction cache which are defined behaviorally; and the cost analysis is performed with the Synopsys synthesis tools using LSI Logic's lca500kv technology.

4.1 Properties of the Modeled Pipeline

The in-order pipeline that is designed in this work has five stages, namely Instruction Fetch (IF), Instruction Decode (ID), Instruction Execute (EX), Memory Access (MEM) and Write Back (WB) stages. The instruction and data caches that are used are 64KB direct mapped caches. For simplicity, no other memory hierarchy is modeled.

A simple data forwarding path which forwards the results from MEM and WB stages to the EX stage is used so that there are no stalls due to data hazards. Figure 4.1 shows the basic properties of the pipeline modeled.

However, a stall condition arises due to load use hazard. If the instruction that follows a load instruction uses the result of the load instruction, then it should be stalled in the ID stage since the result value of the load instruction will be available at the end of its MEM cycle.

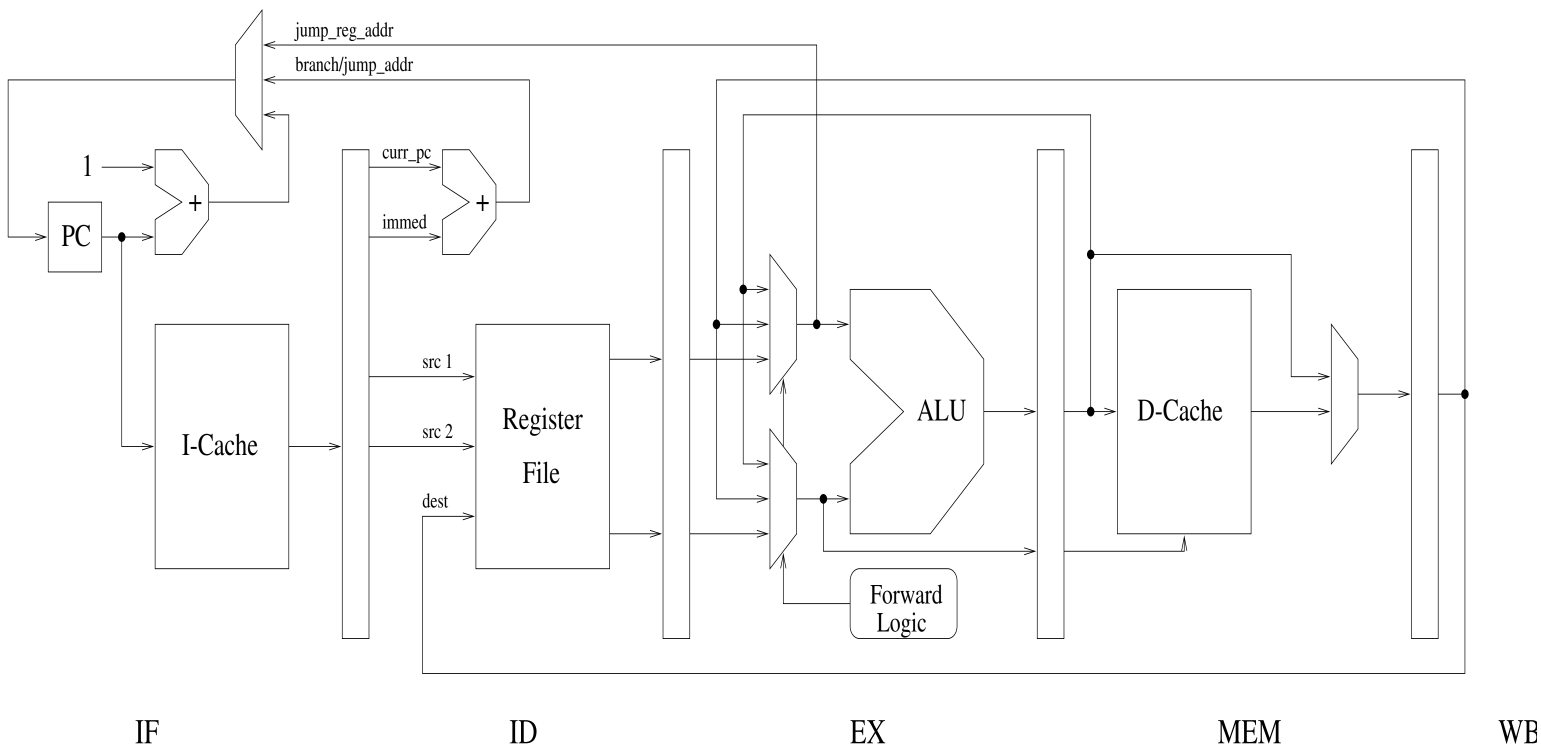

Figure 4.1 Basic structure of the pipeline modeled

The branch instructions in the DLX instruction set compare their operands only with ‘0’, so the branch outcome can be calculated as soon as the source register is fetched in the ID stage. Since there is no forwarding path to the ID stage, in the case of a branch hazard (i.e. when the source register of the branch instruction will be modified by an instruction in EX or MEM stages), the branch instruction is stalled in the ID stage. However, there isn’t a branch hazard when the only instruction modifying the source register of the branch instruction is in the WB stage due to the implementation of the register file, which is modified in the first half and read in the second half of the clock cycle. There is no dynamic branch prediction scheme used in this implementation; instead, branches are statically predicted to be not-taken all the times. The result of this scheme is losing one cycle per a taken branch. No interrupts and exceptions have been modeled.

Since the SRAM portions of the design were not structurally implemented, they could not be synthesized. Instead, the data given in the LCA500K Memories Data Book is used. Both the instruction and the data caches have 16384 entries, 64KB of data and 32KB tag with one read/write port. The register file has 32 32-bit registers and 2 read and 1 write ports. The specifications of these SRAM blocks are given in Table 4.1.

Table 4.1: The SRAM Portions of the Modeled Pipeline

SRAM Type	Read Access Time	Write Access Time	Gate Count
Instruction and Data Caches	13.8 ns	9.38 ns	106920
Register File	3.9 ns	2.9 ns	14136

Each stage of the pipe is synthesized, and the results in Table 4.2 are obtained. The results shown take the read and/or write times of the SRAM portions into account but not the gate counts. In all the tables throughout this thesis, the 'combinational area' is in terms of gate counts (i.e. has the unit of gate count), the 'non-combinational' (sequential) and the 'net interconnect area' are in terms of the gate count of a combinational part with equivalent area; 'total cell area' is the sum of both the combinational and non-combinational area; 'total area' is the sum of the total cell area and the net interconnect area. And the data arrival time is the delay through the corresponding stage including the latches.

Table 4.2: Synthesis Results for Each Stage of the Modeled Pipeline

Property	IF	ID	EX	MEM	WB	Total
Combinational Area	701	431	2690	2	139	3959
Non-Combinational Area	994	1004	648	636	0	3282
Net Interconnect Area	671.74	504.03	2578.01	172.58	115.25	4041.61
Total Cell Area	1695	1435	3338	638	139	7245
Total Area	2366.74	1939.03	5916.01	810.58	254.25	11286.6
Data Arrival Time	16.09 ns	7.8 ns	13.5 ns	13.8 ns	2.53 ns	16.09 ns

These results are obtained for optimum area, in other words, no delay constraints were given to the synthesis tools in order to minimize the area. The bottleneck of these five stages is the IF stage, which is mostly due to the Instruction Cache access, the data arrival time of which suggests a clock rate of approximately 62 MHz. This clock rate seems to be low because of two reasons; the first one being the handling of latches by the synthesis tools, and the second one being the non-pipelined design of the IF stage (both the Instruction and the Data Caches are accessed in one cycle). But these will not affect our cost analysis since this speed and area will be our reference point for hardware cost and critical path analysis.

4.2 The Additions and Modifications to the Hardware for GUARD Support

The first modification that has to be done is introducing the *GUARD* type instructions to the Instruction Decode unit, so that necessary decoding bits are generated for these instructions.

The second modification is adding the simple decoder for the *Mask Field* of the *GUARDBOTH* instruction which was described in Chapter 3. The design of this decoder is described in section 4.2.1.

The third, and the biggest modification is adding a structure to the pipeline which will keep track of the state of the instructions (i.e. whether they are to be executed or to be converted into NOPs). As proposed in [1], the structure that fits this definition is a special kind of a *'Shift Register'*. The design of the shift register used in this work is described in section 4.2.2.

The fourth modification which is a rather straightforward one is to the PC incrementer logic, which is also mentioned in section 4.2.2.

The gate counts, area overheads, timings and the critical path effects of these designs are discussed in section 4.2.3.

4.2.1 Decoder Logic for the Mask Field of GUARDBOTH Instructions

The decoder logic that is designed for the mask field of the *GUARDBOTH* instruction according to the encoding data in Table 3.2 is synthesized using LSI Logic's lca500kv synthesis libraries. The resulting combinational logic diagram with the critical path highlighted is given in Figure 4.2. And the properties of this individual circuit in terms of area, gate count and timing is given in section 4.2.3. This logic is duplicated four times since one block decodes only five bits, and there are 20 bits to decode. Since the earliest time that the Guard Instruction (hence the mask field) is ready is the ID stage, this block is placed as an ID stage component. The results are latched so that they can be used in the EX stage. This introduces a 24 bit latch overhead which is also taken into account in section 4.2.3.

In Figure 4.2, the *mask* output defines whether the corresponding instruction is guarded by the GUARDBOTH instruction or not. A ‘1’ in a specific bit position indicates that the instruction corresponding to that bit position is guarded, and ‘0’ indicates that it is not.The *tf* (true/false) output defines the condition on which the corresponding instruction is guarded. If the instruction is not guarded at all, then this field is a don’t care. The bits are ‘1’ or ‘0’ indicating that the instruction is guarded on the *Guard Condition Register* being ‘true’ and ‘false’ respectively.

4.2.2 Shift Register Design for Support of Guarding

The shift register is needed to keep track of which instructions will be executed, and which will be turned into NOPs. Since this register will be modified only by the *GUARD* instructions which can guard upto a maximum number of 21 following instructions, the register should consist of at least 21 bits. The shift register designed in this work has 21 bits and the least significant bit corresponds to the instruction in the ID stage, the next bit corresponds to the instruction in the IF stage and the rest of the bits correspond to the instructions that are not fetched yet.

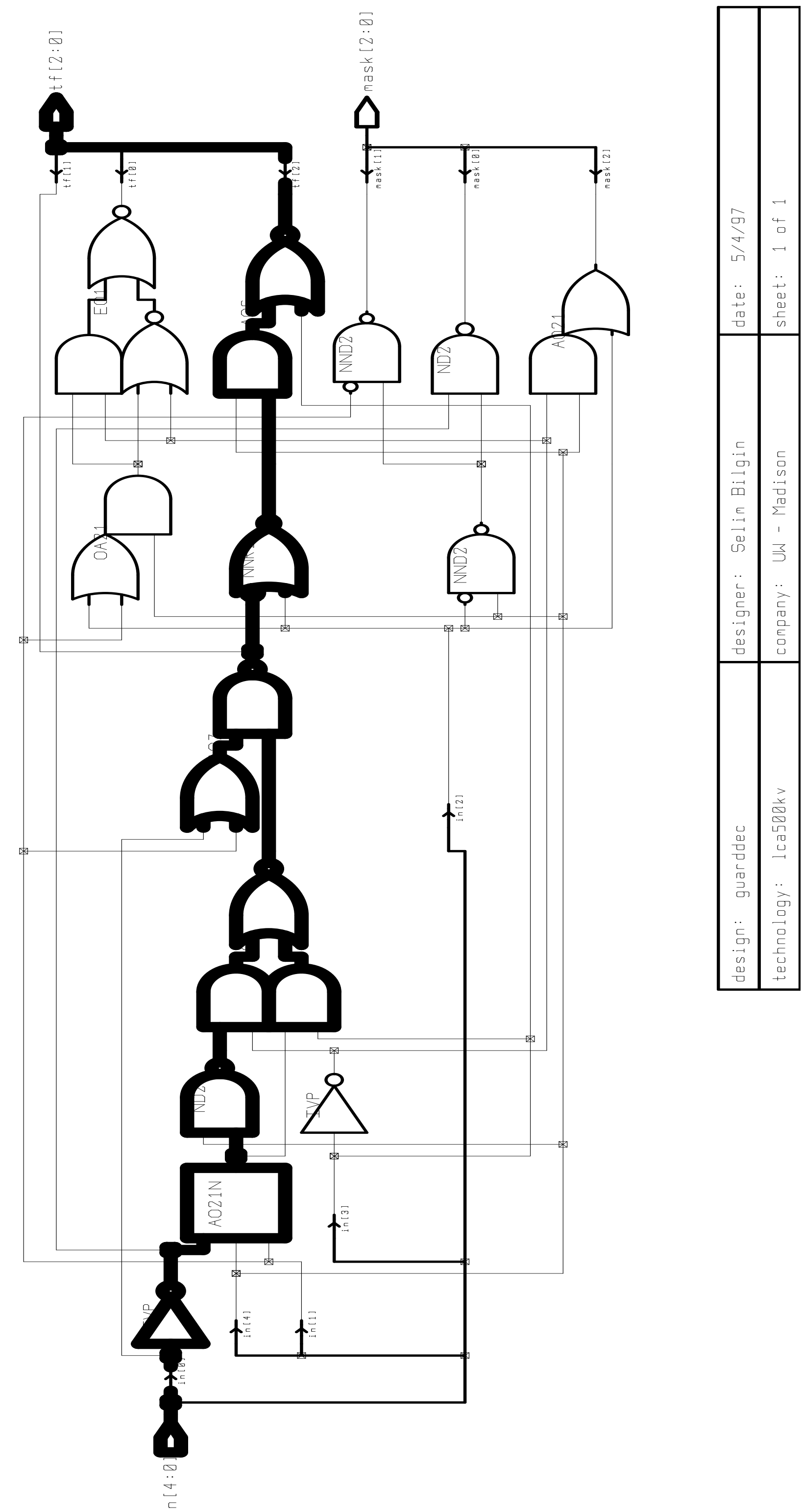

Figure 4.2 The logic and the critical path of the Mask Field Decoder for GUARDBOTH

The shift register works as following; if the corresponding bit of an instruction in the shift register is set (‘1’) then the instruction is a candidate of execution; on the other hand if it is reset (‘0’), then the instruction is definitely to be turned into a NOP. If the shift register has its least significant bit set (‘1’), then the instruction corresponding to that bit (the instruction in the ID stage), is definitely going to be executed.

Resolving the interaction between the *Branch* type and the *GUARD* type instructions is not straightforward because the *GUARD* instructions are based on the assumption that instructions in the region that is guarded are fetched sequentially; but unfortunately taken branches violate this assumption. So, any bits that are ‘0’ (not fetched but squashed instructions) in the shift register after a taken branch will not make sense, since they no more correspond to the same instruction that they did before the branch was taken. The solution used in this design is setting all the bits in the shift register to ‘1’ when a branch is taken regardless of its direction. This solution poses some limits to the code to be executed all of which have been studied in [1].

In order to generate the data that the shift register is going to be modified with, some hardware should be added to the pipeline. The earliest time that the *Guard Condition* can be evaluated is the EX stage since the *Guard Condition Register* will be available at the end of the ID stage. Due to this reason, the hardware will be added to the EX stage. The first thing that has to be added is a 32-bit wide OR gate which decides if the contents of the *Guard Condition Register* is true or false, because according to the first option that is implemented, for a value to be false, all bits of the register containing that value should be 0’s. On the other hand as a second option, only the least significant bit of the register can be checked, and if it is zero, than the value can be said to be false, otherwise true; when this option is chosen for implementation, then the 32-bit OR gate is not needed. This second option is also analyzed in this section.

The second piece of hardware that has to be added is the block which generates the *modify* signal which causes the shift register to be modified with the appropriate data. This *modify* signal will be generated whenever there is a valid *GUARD* type instruction in the EX stage. Just a two input AND gate which have the *ex_valid* (indicating that the instruction in the EX stage is a valid instruction) and *guard_in_ex* (the decode bit that indicates that the instruction in the EX stage is a *GUARD* instruction) as inputs will be appropriate to generate this signal.

The third piece that has to be added to the EX stage forms the data that tells the shift register if the condition holds or not. If the condition register content is false and the following individual instruction is guarded on true condition, or if the register content is true and the following individual instruction is guarded on false condition, then the corresponding bit of this data is '0', otherwise '1'. This data field produced is independent of the mask field of the *GUARD* type instruction. If the instruction is a *GUARDBOTH* instruction then the most significant 9 bits of this data are not used, instead they are hardwired to the value '0'. This data produced is referred to as the *condition field* (*cond*) throughout this work.

The *Mask Field* which is directly sent to the shift register is also produced in this stage, which is rather easy because for a *GUARDTRUE* and a *GUARDFALSE* instruction the *Mask Field* is the lower 21 bits of the instruction, and the *Mask Field* of the *GUARDBOTH* instruction is formed in the ID stage and latched for use in the EX stage. The circuit diagram of these pieces of hardware with the critical path highlighted is given in Figure 4.3. And Figure 4.4 shows the circuit diagram of the same logic when only the least significant bit is used for determining whether the value is true or false. And the properties of these circuits in terms of area, individual gate count and timing is given in section 4.2.3. As can be seen from Figure 4.3 the 32-bit

OR gate is on the critical path of this piece. When the OR gate is removed and only the least significant bit is tested, the critical path remains the same except that the OR gate is obviously not on the critical path anymore.

The modification of the shift register data requires careful action. First, the data that the register is going to be modified with should be prepared. If the shift register is not signaled to be modified, then this data should be the same as the shift register contents. If there is a modification, then the entries in the original shift register contents which are ‘0’ should be left as ‘0’; and the entries that are ‘1’ should be turned into a ‘0’ if and only if the corresponding instruction is masked by the *GUARD* instruction which is currently in the EX stage and the condition of which is evaluated such that the instruction will be turned into a NOP. The equation for this relationship is:

$$\text{new\_data}_i = (\sim\text{modify} \;\&\; \text{register\_bit}_i) \mid (\text{modify} \;\&\; \text{register\_bit}_i \;\&\; \sim(\text{mask}_i \;\&\; \sim\text{cond}_i))$$

As mentioned above, it may be the case that some not yet fetched instructions have been turned into NOPs; so there is no point in fetching such instructions. In other words, whenever an instruction is going to be fetched, it should not have been turned into a NOP. This is done by counting the number of instructions starting from the one which is in bit position 2 (the first not yet fetched instruction, the least significant two bits correspond to the instructions already in the IF and ID stages which are already fetched) which have a ‘0’ in the corresponding bit of new_data bus. Then the ‘*count*’ of instructions that have been turned into NOPs is sent to the PC incrementer logic which will increment the PC by 1+’*count*’. The PC incrementer is also modified so that it can increment by ‘1+’*count*’.

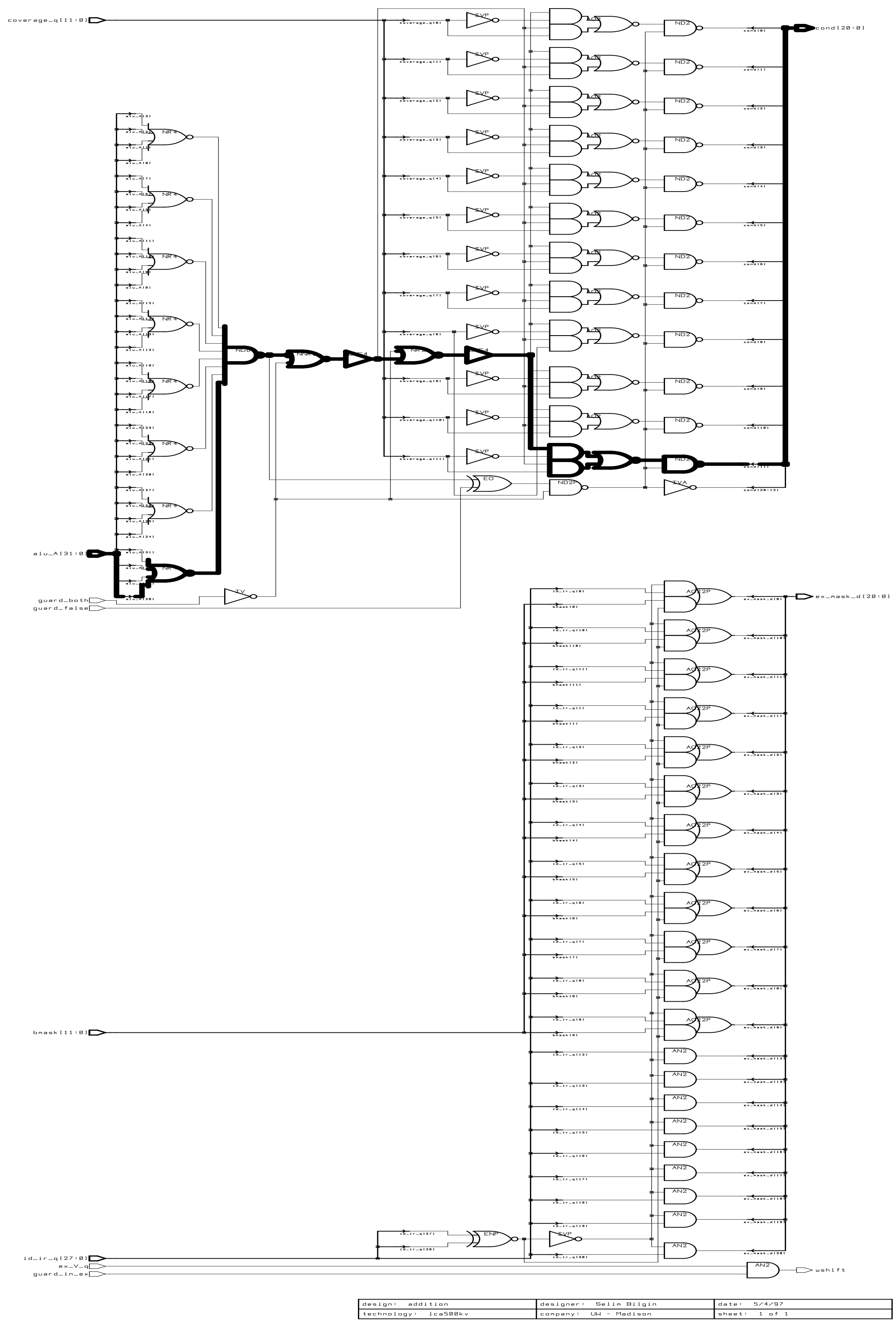

Figure 4.3 The logic diagram and the critical path of the hardware added to the EX stage

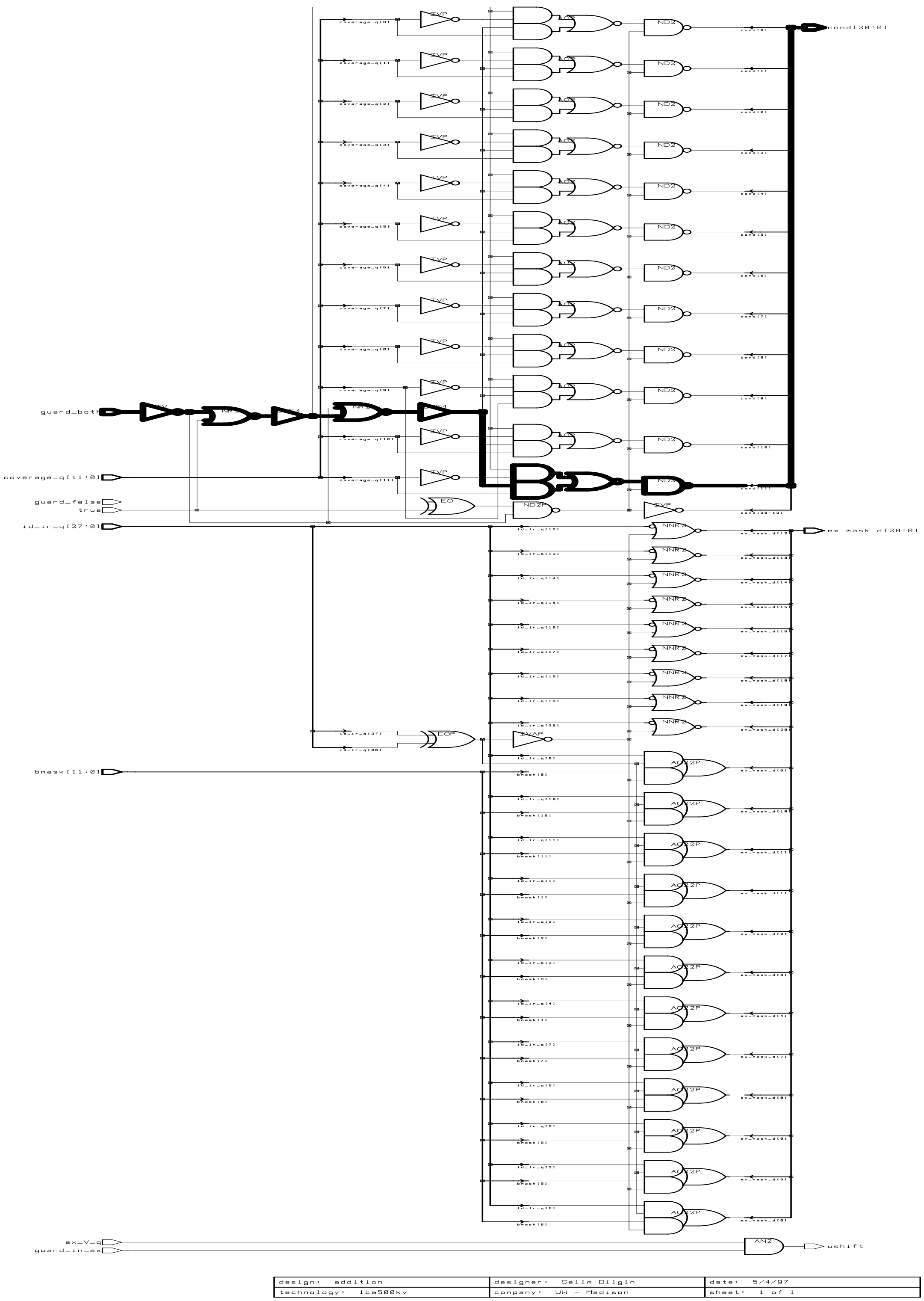

Figure 4.4 Same as Figure 4.3., without the 32-bit OR gate

The sub-logic that is used in figuring this '*count*' out is shown in Figure 4.5. It takes 4 bits as input and produces a '1' at the position in which the first '1' is seen starting from the least significant bit of the input, leaving both the higher and the lower order bits as '0'. Five of these blocks are connected in a carry look ahead fashion such that if a '1' is seen at the output of a block, then the outputs of the higher order blocks are disabled (all reset to '0'). If no '1' is seen, then the count is set to 19, which is the number of yet unfetched instructions which have an entry in the shift register. The hardware overhead of the 'one's position detector' logic is included in the investigation of the overhead of the 'shift register'.

If there is a stall, then the shift register is modified directly with '*new_data*'. If there is no stall, but a taken branch, then the contents of the shift register are all set to '1'. If there is neither a stall nor a taken branch, then the bit position 0 of the register is modified with bit position '1' of '*new_data*', the bit position 1 of the shift register is assigned to '1', and the rest is modified with the bit positions [20:2] of '*new_data*' shifted '*count*' times right (towards the LSB), and its unassigned portion (due to shifting) assigned to '1', since those instructions were not covered by any *GUARD* instruction.

The interface of the shift register is shown in Figure 4.6. And the circuit diagram for the shift register and its controller logic after synthesis is shown in Figure 4.7. The properties of this circuit in terms of area, individual gate counts and timing is given in section 4.2.3. The meanings of the abbreviations used in Figure 4.6 are: *cond[20:0]* input is the '*data*' bus produced in the EX stage. The *data_in[20:0]* input is the actual mask field of the *GUARD* instructions. *count[4:0]* is the number of instructions to be skipped by the fetch unit, that is sent to the PC incrementer logic. The *data_d[20:0]* is the actual shift register contents which are sent to the pipeline.

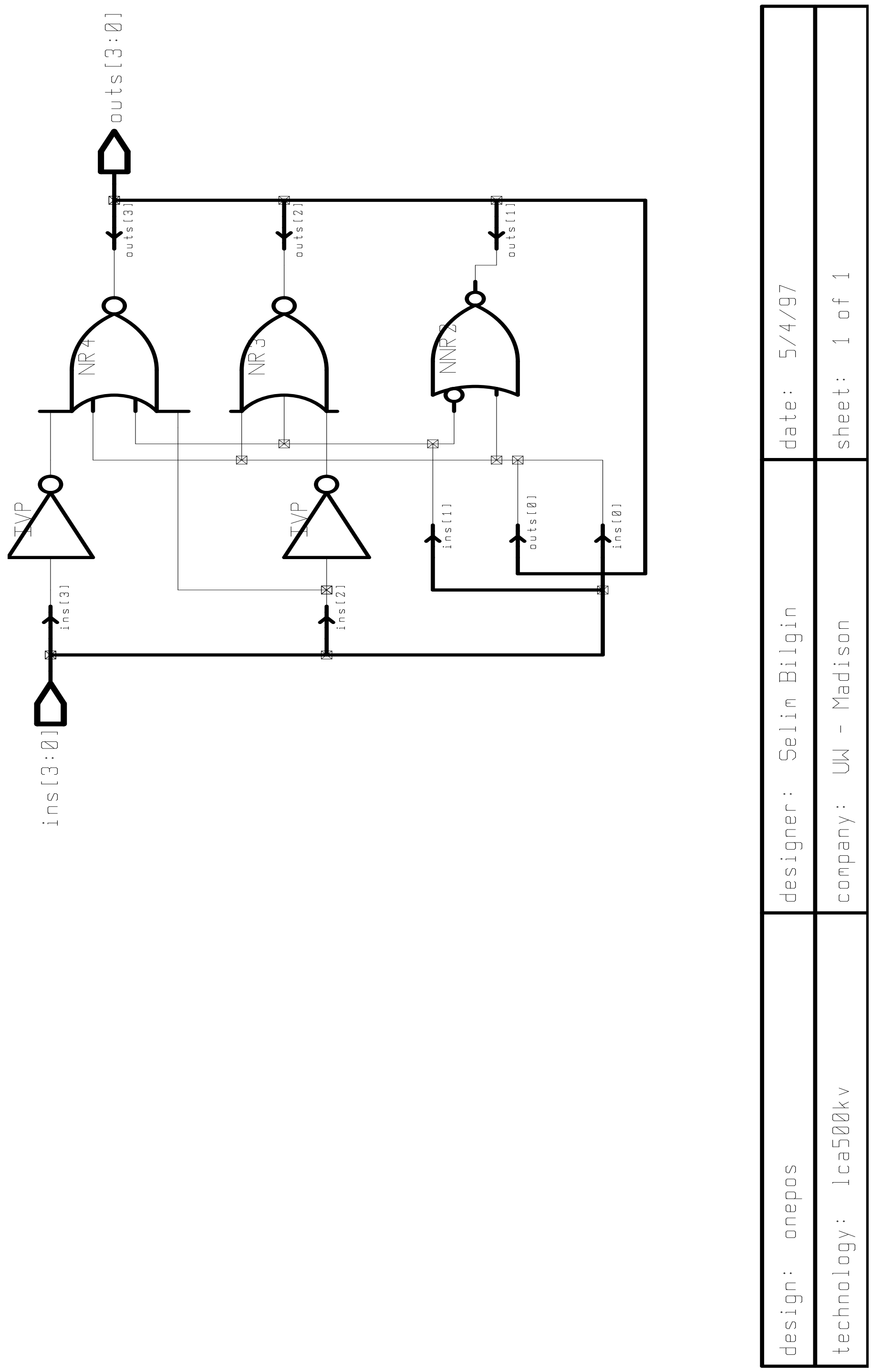

Figure 4.5 The circuit diagram of the "One's Position Detector Logic"

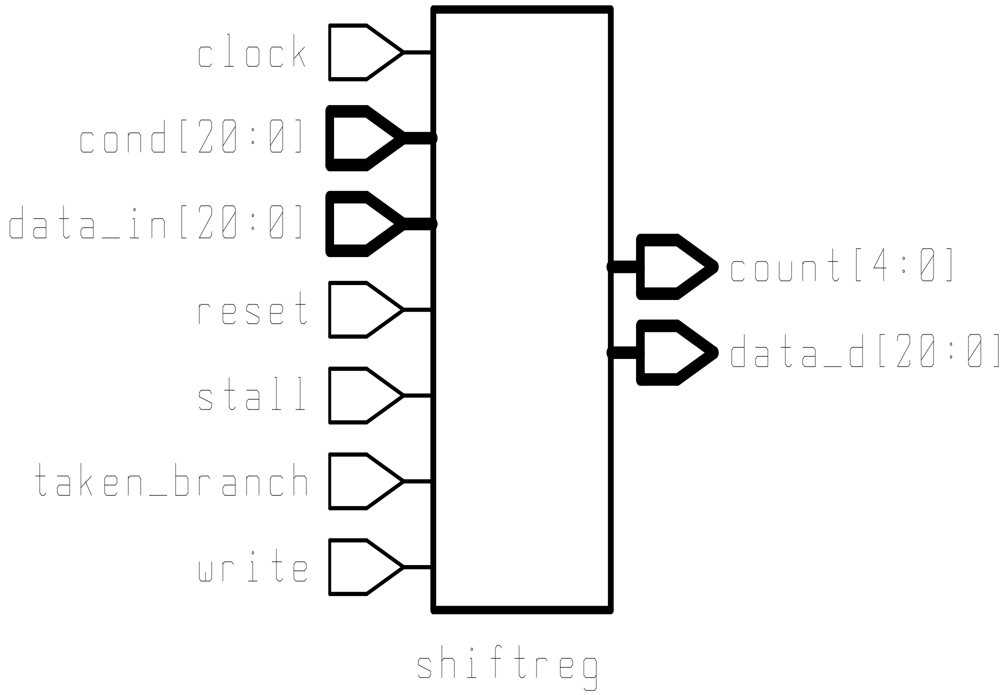

Figure 4.6 The interface to the Shift Register

4.2.3 Area, Gate Count and Timing Overheads

In this section, the hardware overhead of implementing *Guarded Execution* will be investigated. The base case is given in Table 4.2. Table 4.3 shows the modified case when optimized for minimum area. The areas reported are again in terms of equivalent gate counts. The delay through the IF and ID stages do not change due to the SRAM timing constraints. The MEM and WB stages do not change since the logic in those stages are not modified. The only stage that timing of which is affected is the EX stage. The hardware, synthesis results of which is shown in Table 4.3, tests only the least significant bit of the condition register.

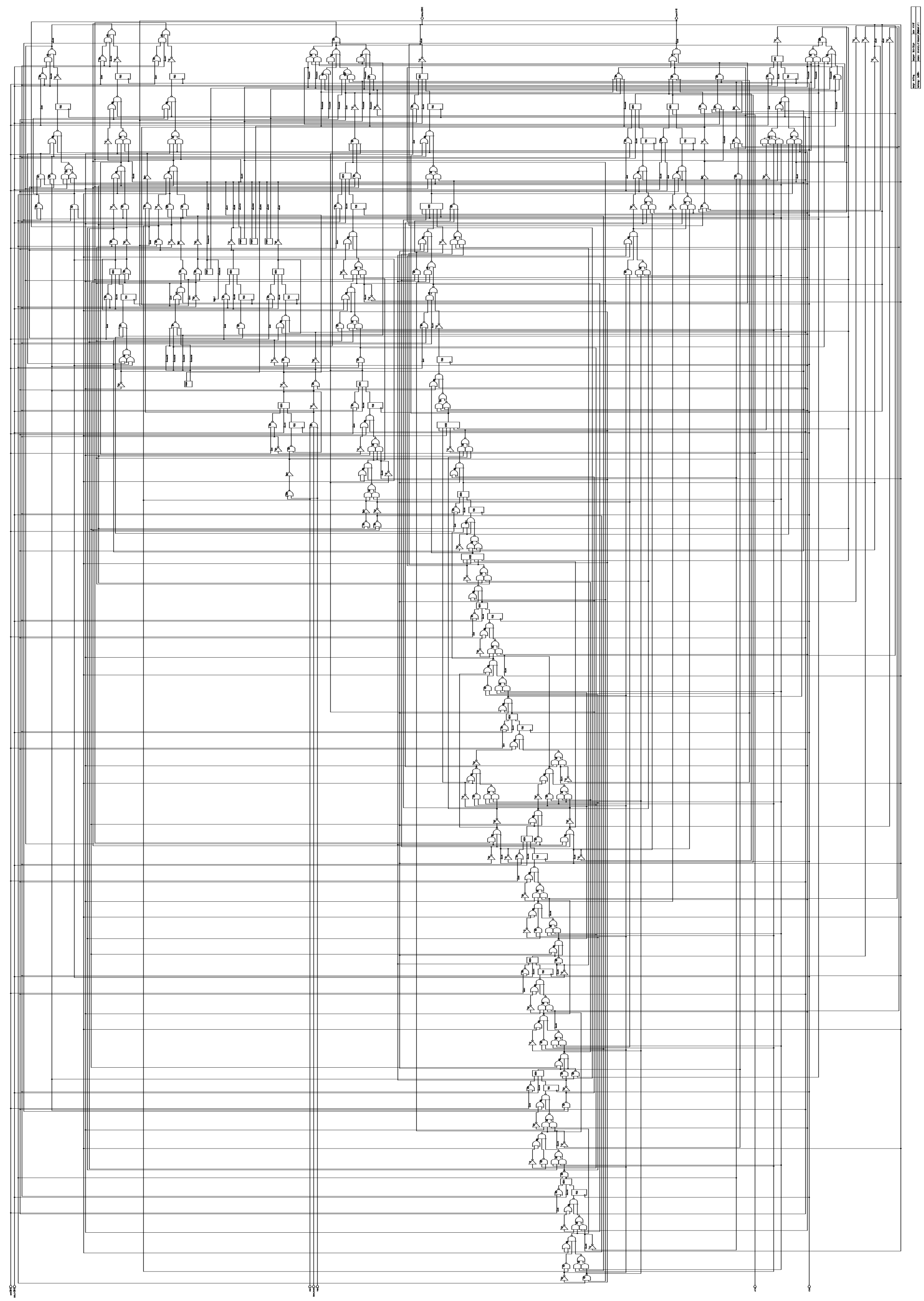

Figure 4.7 The circuit diagram of the Shift Register

Table 4.3: Modified pipeline synthesis results for minimum area

Property	IF	ID	EX	MEM	WB	Total
Combinational Area	857	547	2995	2	139	4540
Non-Combinational Area	994	1260	858	636	0	3748
Net Interconnect Area	703.70	752.62	2916.28	172.58	115.25	4660.43
Total Cell Area	1851	1807	3853	638	139	8288
Total Area	2554.70	2559.62	6769.28	810.58	254.25	12948.4
Data Arrival Time	16.09 ns	7.8 ns	13.65 ns	13.8 ns	2.53 ns	16.09 ns

The percent differences between Table 4.3 and Table 4.2 are given in Table 4.7. In the remaining part of this section, the individual synthesis results of the hardware pieces added are given. The gate types in the synthesis results are described in Appendix 1.

The first part of hardware that is investigated is the Decoder logic for the mask field of the *GUARDBOTH* instructions. The gate counts and the area occupied are given in Table 4.4.

Table 4.4: Gate types, counts and the area occupied by the Mask Decoder

Gate Type	Unit Area	Count	Total Area
AO2	2	1	2
AO6	2	1	2
AO7	2	1	2
AO21	2	1	2
AO21N	2	1	2
EO1	3	1	3
IVP	1	2	1

Table 4.4: Gate types, counts and the area occupied by the Mask Decoder

Gate Type	Unit Area	Count	Total Area
ND2	1	2	2
NND2	2	2	4
NNR2	2	1	2
OA21	2	1	2
Latches	6	6	36
Net Interconnect	N/A	N/A	20.11
Total	**27**	**20**	**80.11**
Max	**6**	**6**	**36**

One block of the decoder occupies an area of 80.11, since four of these decoders are needed, then the total area required is 320.44. The latches involved in this design are used for latching the data to be used in the EX stage. The delay through the critical path of one single decoder is 1.32 ns.

The second part of the hardware that was investigated in previous sections is the additional hardware required in the EX stage for evaluating the condition of *guarding*, for forming the condition field of the *GUARD* instructions, and for generating the *modify* signal for the shift register. The gate counts and the area occupied by these pieces of hardware is given in Table 4.5.

Table 4.5: Gate types, counts and the area occupied by the added hardware to EX stage

Gate Type	Unit Area	Count	Total Area
AN2	2	10	20
AO2	2	12	24
AO22P	3	12	36

Table 4.5: Gate types, counts and the area occupied by the added hardware to EX stage

Gate Type	Unit Area	Count	Total Area
BUF4	2	2	4
ENP	4	1	4
EO	3	1	3
IV	1	1	1
IVA	1	1	1
IVP	1	13	13
ND2	1	12	12
ND2P	2	1	2
ND8	6	1	6
NNR2	2	1	2
NR2	1	1	1
NR4	2	8	16
Net Interconnect	N/A	N/A	122.60
Total	**33**	**77**	**267.60**
Max	**6**	**13**	**36**

The total area overhead of this piece of hardware is 267.60. As can be seen from Table 4.5, there are no latches involved in this design, because the data which forms the input to this piece of logic is fed from the latches which were already present in the original pipeline.

The delay through the critical path of this logic is 2.61 ns. This number is of interest to us because this logic and the shift register form the total path that is added to the EX stage. If the scheme in which only the least significant bit is checked for deciding on the condition register being true or not is implemented, then the area occupied reduces to 220.45 while the delay through the critical path reduces to 1.96 ns.

The gate counts, and the area occupied by the shift register is given in Table 4.6.

Table 4.6: Gate types, counts and the area occupied by the Shift Register

Gate Type	Unit Area	Count	Total Area
AN2P	2	2	4
AN3	2	2	4
AO1	2	5	10
AO2	2	29	58
AO3	2	36	72
AO4	2	2	4
AO6	2	5	10
AO7	2	1	2
AO7P	3	1	3
AO11	5	1	5
AO21	2	1	2
AO21N	2	17	34
AO22P	3	1	3
AO211	3	1	3
AO222	4	1	4
BUF1	1	2	2
BUF4	2	4	8
EO1	3	2	6
FD1Q	6	21	126
IV	1	36	36
IVA	1	22	22
IVP	1	4	4
ND2	1	27	27
ND2P	2	2	4
ND3	2	2	4

Table 4.6: Gate types, counts and the area occupied by the Shift Register

Gate Type	**Unit Area**	**Count**	**Total Area**
ND3P	3	1	3
ND4	2	2	4
NNR2	2	2	4
NR2	1	29	29
NR2P	2	1	2
NR4	2	4	8
OA222	4	2	8
OR3	2	2	4
OR4	3	1	3
onepos	8	5	40
Net Interconnect	N/A	N/A	412.44
Total	**87**	**276**	**974.44**
Max	**8**	**36**	**126**

The above analysis includes five copies of the ‘One’s Position Detector” blocks. The delay through the critical path of this circuit is 5.80 ns.

The final modification is to the PC incrementer logic that is used in the IF stage for calculating the address of the next to be fetched instruction. The synthesized version of this block occupies an area of 329.42, and has a delay of 3.77 ns. The synthesis output reports are given in Appendix 2.

4.3 Conclusions

In this chapter the hardware that is to be added to the in-order pipelined processor for supporting *Guarded Execution* was investigated. In this section, the numbers that are obtained are analyzed.

When the data in Table 4.3 is compared with the data in Table 4.2, the results in Table 4.7 are obtained.

Table 4.7: Percent overhead of GUARDed Execution for each stage

% increase	IF	ID	EX	MEM	WB	Total
Combinational Area	22.25%	26.91%	11.34%	0%	0%	14.68%
Non-Combinational Area	0%	25.50%	32.41%	0%	0%	14.20%
Net Interconnect Area	4.76%	49.32%	13.12%	0%	0%	15.31%
Total Cell Area	9.2%	25.92%	15.43%	0%	0%	14.40%
Total Area	7.94%	32.01%	14.42%	0%	0%	14.72%
Data Arrival Time	0%	0%	1.11%	0%	0%	0%

Since the access time for the Instruction Cache is high when compared to the rest of the logic in the IF stage, the new PC incrementer does not affect the critical path delay of the IF stage. Also, since the access time for the register file is 3.9 ns, and it can be read only in the second half of the cycle, it introduces a 7.8 ns delay; and the delay of the Mask Field Decoder for GUARDBOTH instructions which is placed in the ID stage is only 1.32 ns, therefore it has no effect on the critical path delay of the ID stage either.

When the worst case delay of the shift register and the delay of the hardware added to the EX stage are combined, the resulting worst case delay is 8.41 ns. This delay is still far below 13.5 ns which is the critical path delay of the EX stage. But when combined with the original hardware and optimized for minimum area, this number has an effect on the critical path delay of the EX stage.

However, there is one small limitation of the design presented in this chapter, that is the sequentially next two instructions after a *Guard* type instruction can not be guarded by a previous *Guard* type instruction. That is because if they are, then just after the second *Guard* instruction is fetched, the two others will be squashed and will never be fetched, hence the second *Guard* instruction will be guarding some wrong instructions.

Chapter 5

Integrating GUARD Instructions into an Out-of-Order Pipelined Architecture

In this chapter, the costs of integrating the three *GUARD* type instructions into an existing out-of-order pipelined architecture is investigated. The modeled machine is capable of executing the instructions given in Table 3.1. The pipeline is structurally implemented using Verilog HDL except its memory components such as register file, the Branch Target Buffer, the Branch History Table, the data cache and the instruction cache which are defined behaviorally as in the in-order case; and the cost analysis is performed with the Synopsys synthesis tools using LSI Logic's lca500kv technology.

5.1 Properties of the Modeled Pipeline

The out-of-order pipeline that is designed in this work has six stages, namely Instruction Fetch (IF), Instruction Decode (ID), Instruction Issue (IS), Instruction Execute (EX), Memory Access (MEM) and Write Back (WB) stages. Although all these six stages are explicitly named here, in order to keep the architecture similar to the MIPS architectures, especially the state of the art MIPS R10000 processor [4], not all the instructions go through all the stages. (i.e. non memory accessing instructions need not and do not go through the MEM stage). The general block structure of data flow is shown in Figure 5.1. The data going to the EX and MEM stages are latched, but the data going to the WB stage is not latched since it is already coming out of a latch in the IS stage (ROB and Reservation Station).

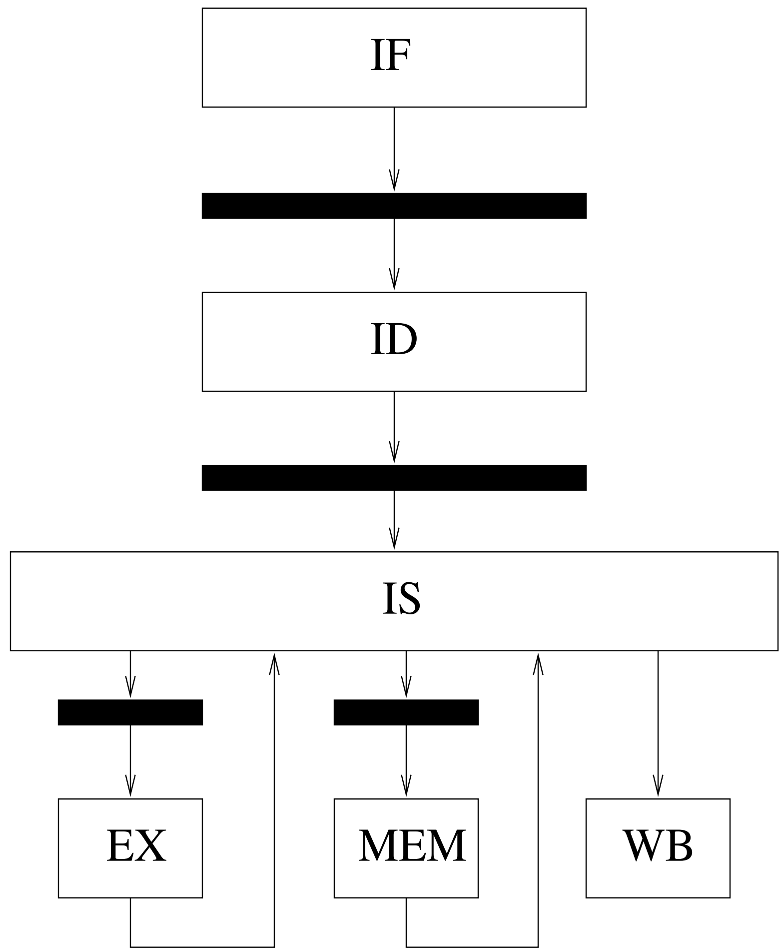

Figure 5.1 Data flow diagram of the out-of-order pipelined design

The instruction and the data caches that are used are 64KB direct mapped caches. As in the in-order case, again for simplicity, no other memory hierarchy is modeled.

The processor modeled is able to issue one instruction per cycle, and it has an 8-entry Reorder Buffer (ROB). In addition to the Reorder Buffer, there is an 8-entry Reservation Station modeled for issuing the instructions to the ALU and the data cache. Also, 4-entry Load and Store queues are employed. In the processor modeled, a Tomasulo like scheme is maintained with the above properties.

The issue process stalls whenever the ROB is full, or the Load Buffer is full and the next instruction to be issued is a Load type instruction, or the Store Buffer is full and the next instruction to be issued is a Store instruction. When an instruction is issued, an entry is allocated to it in both the ROB and the Reservation Station; also an entry is allocated for it in the Load Buffer or the Store Buffer if it is a memory referencing instruction. The instruction which is the head of the ROB has priority to advance to the EX and MEM stages; in the case that the instruction which is the head has already finished execution but not committed yet, then the instructions that are going to advance to the EX and MEM stages are determined statically; giving the static priority to the lower numbered ROB stages. This scheme reduces the complexity of selecting the instruction to be executed next. The load instructions can not access memory whenever there is an address hazard, in other words no load speculation is employed; all the load instructions wait for the preceding stores which have their address undetermined or the same as the load instruction.

A 2-bit Branch History Table is used along with a Branch Target Buffer (BTB) for branch prediction. The number of entries in the BTB and the Branch History Table is 1024. Whenever the head of the ROB is a mispredicted branch, then the whole ROB is squashed and the execution is resumed by fetching from the correct PC. Along with branch prediction, jump prediction is also implemented.

Since the SRAM portions of the design were not structurally implemented, they could not be synthesized. Instead, the data given in the LCA500K Memories Data Book is used. The instruction and data caches have the same properties as in the in-order case. The Branch Target Buffer has 1024 entries and one read and one write port. Total data stored in the BTB is 3.75KB, and the tag space is 2.5 KB. The branch history table has also 1024 entries and one read and one write port. The data stored in the branch history table is 0.25KB. The specifications of these SRAM blocks are given in Table 5.1. (Nearest values available for the BTB and the Branch History Table are used). The access times of the BTB is comparable to the access times of the instruction and the data caches since it is two ported instead of one.

Table 5.1: The SRAM Portions of the Modeled Pipeline

SRAM type	Read Access Time	Write Access Time	Gate Count
Instruction and Data Caches	13.8 ns	9.38 ns	106920
Register File	3.9 ns	2.9 ns	14136
BTB	10.3 ns	9.1 ns	61608
Branch History Table	3.7 ns	4.1 ns	3216

Each stage of the pipe is synthesized, and the results in Table 5.2 are obtained. As in Chapter 4, the results shown take the read and/or write times of the SRAM portions into account, but not the gate counts. The combinational area is in terms of gate counts, the non-combinational (sequential) and the net interconnect areas are in terms of the gate count of a combinational logic with equivalent area; and the arrival time is the delay through the corresponding stage. Since all the control logic for the WB stage is included in the IS stage, and since the register file is considered as a part of the ID stage, WB stage is virtually non-existent in this analysis. The numbers for the IF stage include the latches between IF and ID stages. ID stage has no sequential components (latches) because the latches between the ID and the IS stages are taken into account in the IS stage. According to the results in Table 5.2, 89.3% of the whole design consists of the hardware in the IS stage.

Table 5.2: Synthesis Results for Each Stage of the Modeled Pipeline

Property	IF	ID	IS	EX	MEM	Total
Combinational Area	502	333	35045	2136	0	38016
Non-Combinational Area	1226	0	10908	420	390	12944
Net Interconnect Area	623.70	248.15	20096.01	1925.94	111.82	23005.60
Total Cell Area	1728	333	45953	2556	390	50960
Total Area	2351.70	581.15	66049.01	4481.94	501.82	73965.60
Data Arrival Time	14.54 ns	9.27 ns	19.82 ns	10.15 ns	13.80 ns	19.82 ns

5.2 The Additions and Modifications to the Hardware for GUARD Support

We have investigated the hardware modifications necessary for supporting *GUARD* type instructions in an in-order pipeline in section 4.2. In an out-of-order pipeline, the situation gets a little more complex. The reasons for this complexity are the following; first of all, the *Guard Condition Register* contents may not be available at the time of issue, and secondly, in the worst case, the 7 sequentially following instructions might have been already issued to the ROB, and it might be the case that some or all of them will be converted to NOPs, this results in squashing more than 2 already fetched instructions which was the case for the in-order pipeline. These issues result in the need of storing the mask field in either the manipulated or the un-manipulated form. Also, there are 4 different cases that should be handled; there can be a stall issue, and there can be a commit situation, all the combinations result in 4 different states, and for each different state, the pointers should be updated accordingly. The range for the shift register that is used for keeping track of the states of the instructions should also be increased to 29 since the range of the *GUARD* instructions is 21 and the ROB has 8 entries. The least significant 8 bits of the shift register is statically assigned to the ROB entries; if the tail of the ROB is a *GUARD* type instruction and is the eighth valid entry in the ROB, it will be guarding upto 21 sequentially next instructions, hence it will guard upto the 29th bit of the shift register.

The operation is as follows: when a *GUARD* type instruction is issued to the ROB, the *Guard Condition Register* availability is checked, if it is not available, then the mask of the instruction is stored in the ROB, on the other hand if it is available, then the mask is manipulated depending on the condition being true or not, and the final condition field is stored in the ROB. The mask field of the *GUARDBOTH* instruction is decoded just like it is decoded for the in-order case. The *GUARD* type instructions are resolved before they reach the head of the buffer but they are not executed until they are the head; this is to avoid updating the shift register speculatively because the ROB may get squashed in the case of a branch misprediction. Some ‘busy’ bit is kept for all the instructions that are already fetched and issued to the ROB, the reason being that the instructions which may be converted to NOPs should not be sent to the EX stage when its their turn, because if they will be converted to NOPs, then some execution cycles will be wasted. In addition to the pieces that are described in the next sections, the PC incrementer logic and the decode logic should also be modified. The PC incrementer logic will go through the same design change that it went in Chapter 4.

5.2.1 Value Field Generator Logic

The first piece of logic that is to be added to the pipeline is the one which generates the value fields of the *GUARD* type instructions. The value field of a *GUARD* type instruction is generated depending on its kind and the availability of the *Guard Condition Register.* If the register is not available at the time of the issue, then the value field is basically the mask field that comes with the instruction binary; if the register is available, then the value field is generated depending on the condition being true or false. For the *GUARDBOTH* instructions, the mask field is first decoded, then depending on the availability of the condition register at issue time, it is updated.

The synthesis results of this block is given in section 5.2.4. Only one copy of this logic is used in the entire design. The resulting combinational logic diagram with the critical path highlighted is given in Figure 5.2.

5.2.2 Control Logic For GUARD Instructions

The control logic for the *GUARD* type instructions basically evaluate the condition for guarding, then form the value field out of the mask fields of the *GUARD* instructions which had their condition registers unavailable at the time of issue. The control logic also determines the instructions that are 'busy', in other words the instructions already in the ROB but not yet determined to be executed. Depending on the 4 different combinations of stalling and committing of the instructions, the pointers are also modified by the control logic. (i.e. since the ROB is circular and dynamic, the bits corresponding to each instruction should be modified whenever the state of the ROB changes). The *condition* is determined just by the least significant bit of the *condition register.*

The synthesis results of this block is given in section 5.2.4. For each ROB entry, one copy of this block is used; hence for this design total of 8 copies of this block are used. The resulting logic diagram with the critical path highlighted is given in Figure 5.3.

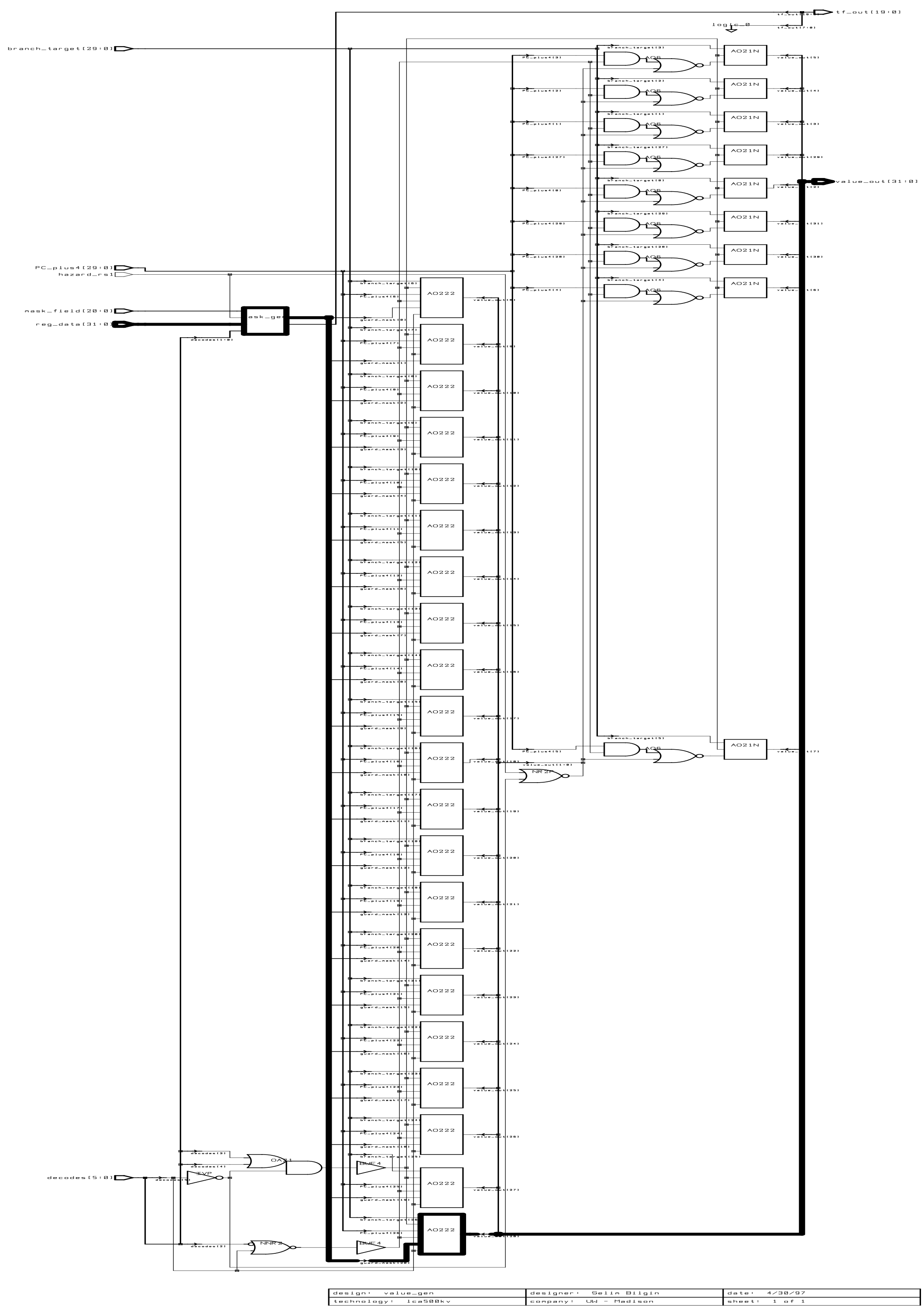

Figure 5.2 The logic and the critical path of the Value Field Generator Logic

5.2.3 The Shift Register

The shift register is used for keeping track of the instructions in terms of whether they are going to be executed or not. The operation of the shift register is similar to the one that was used in the in-order pipeline. However, as mentioned previously, for the out-of-order case, the length of the shift register is increased from 21 to 29. The first 8 entries in the shift register correspond to the instructions in the ROB statically (i.e. no matter how many instructions are valid in the ROB, 8 entries will be allocated in the shift register). The rest 21 entries work in a similar way as the in-order implementation does. The bits that correspond to the instructions in the ROB are manipulated according to the state change of the ROB. (i.e. the commit and stall combinations).

The synthesis results of this block is given in section 5.2.4. Only one copy of this block is used in the entire design. The resulting logic diagram with the critical path highlighted is given in Figure 5.4.

5.2.4 Area, Gate Count and Timing Overheads

In this section, the hardware overhead of implementing *Guarded Execution* in an existing out-of-order pipeline will be investigated. The base case is given in Table 5.2. Table 5.3 shows the modified case when optimized for minimum area. The results are discussed in section 5.3.

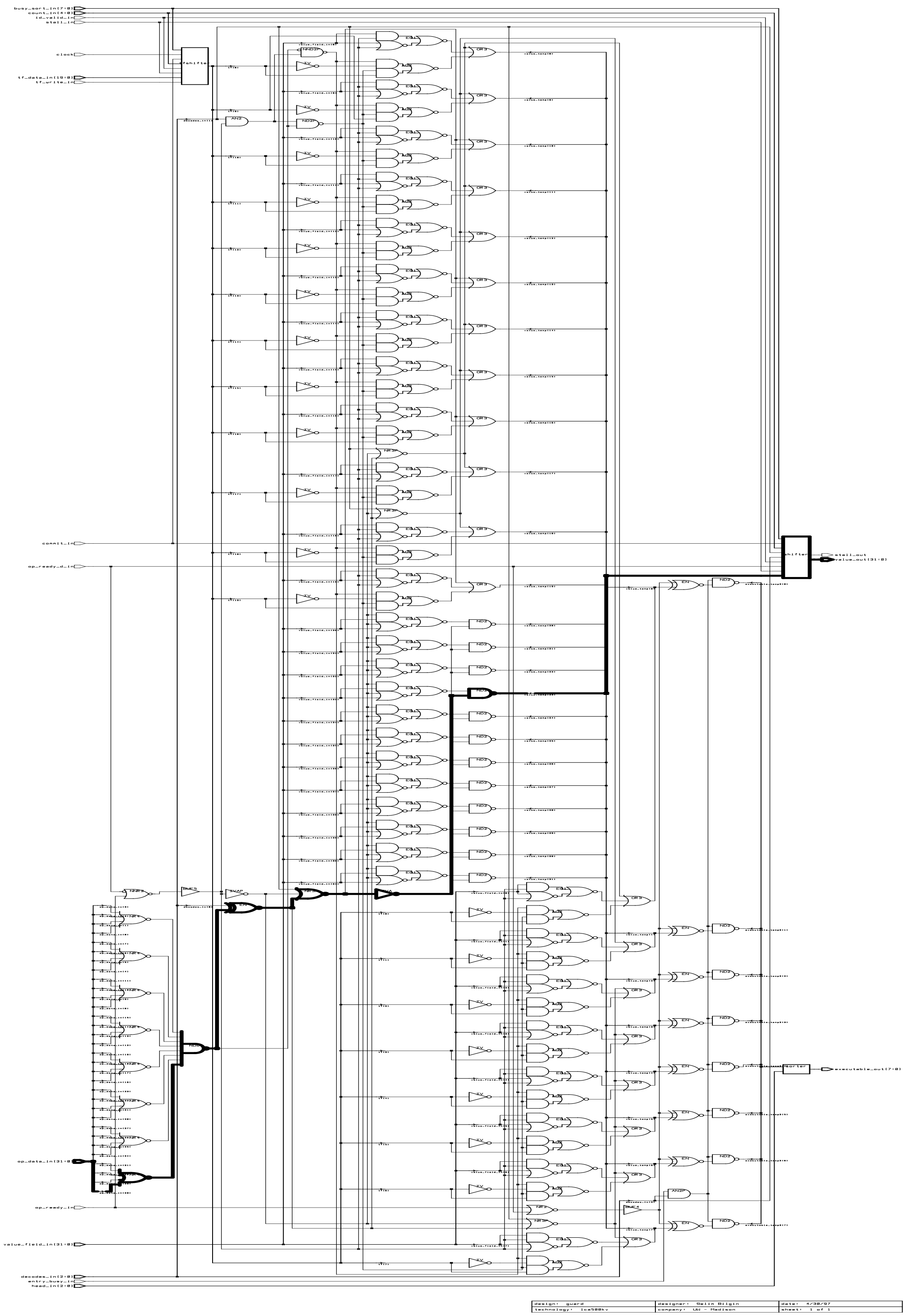

Figure 5.3 The logic and the critical path of the Control Logic for GUARD Instructions

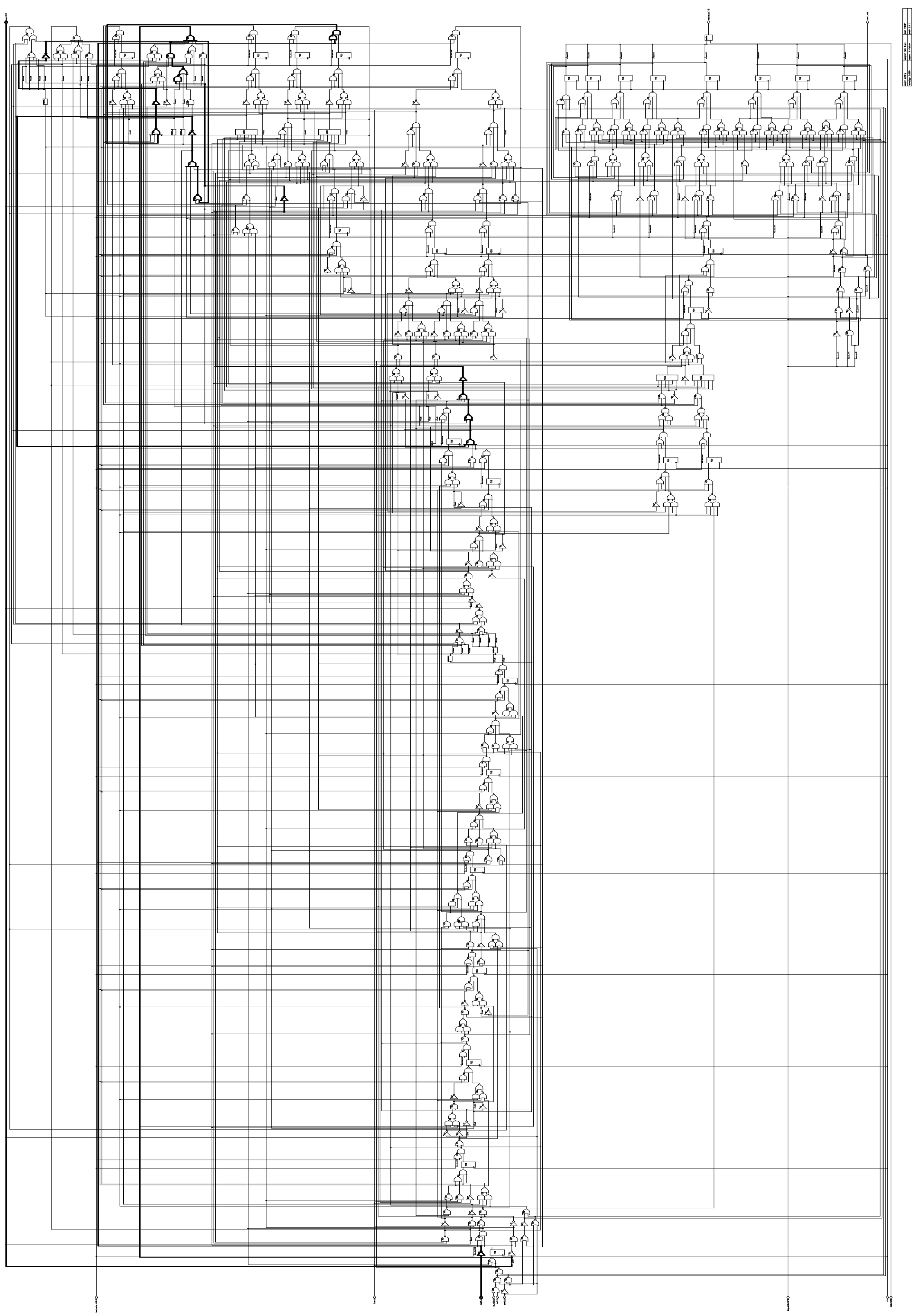

Figure 5.4 The logic and the critical path of the Shift Register

Table 5.3: Modified pipeline synthesis results for minimum area

Property	IF	ID	IS	EX	MEM	Total
Combinational Area	776	344	44245	2136	0	47501
Non-Combina-tional Area	738	0	11082	420	390	12630
Net Interconnect Area	645.86	256.49	21446.50	1925.94	111.82	24386.61
Total Cell Area	1514	344	55327	2556	390	60131
Total Area	2159.86	600.49	76773.50	4481.94	501.82	84517.61
Data Arrival Time	14.95 ns	9.27 ns	20.06 ns	10.15 ns	13.80 ns	20.06 ns

The percent differences between Table 5.2 and Table 5.3 are given in Table 5.7. In the remaining part of this section, the individual synthesis results of the hardware pieces that are added are given.

The first part of the hardware that is investigated is the *Value Field Generator*. The gate counts and the area occupied are given in Table 5.4.

Table 5.4: Gate types, counts and the area occupied by the Value Field Generator

Gate Type	Unit Area	Count	Total Area
AO6	2	9	18
AO21N	2	9	18
AO222	4	21	84
BUF4	2	2	4
IVP	1	1	1

Table 5.4: Gate types, counts and the area occupied by the Value Field Generator

Gate Type	Unit Area	Count	Total Area
NNR2	2	1	2
NR2P	2	1	2
OA21	2	1	2
mask_gen	323	1	323
Net Interconnect	N/A	N/A	360.84
Total	**340**	**46**	**814.84**
Max	**323**	**21**	**323**

One block of the *Value Field Generator* logic occupies an area of 814.84 and the delay through the critical path of this block is 3.36 ns.

The second part of the hardware that is investigated is the *Control Logic for the GUARD Instructions*. The gate counts and the area occupied is given in Table 5.5.

Table 5.5: Gate types, counts and the area occupied by the Control Logic

Gate Type	Unit Area	Count	Total Area
AN2	2	1	2
AN2P	2	1	2
AO2	2	20	40
BUF4	2	1	2
BUF5	3	1	3
EN	3	9	27
EO1	3	32	96
IV	1	20	20
IVA	1	1	1

Table 5.5: Gate types, counts and the area occupied by the Control Logic

Gate Type	Unit Area	Count	Total Area
IVAP	2	1	2
ND2	1	20	20
ND2P	2	1	2
ND8	6	1	6
NND2P	3	1	3
NNR2	2	1	2
NR2	1	1	1
NR3P	3	4	12
NR4	2	8	16
OR3	2	20	40
shifter	413	1	413
sorter	102	1	102
tfshifter	370	1	370
Net Interconnect	N/A	N/A	948.42
Total	**928**	**147**	**2130.42**
Max	**413**	**32**	**413**

One block of the control logic occupies an area of 2130.42, and 8 different copies of this block is used in the design. The delay through the critical path of one of these blocks is 5.94 ns.

The third and the most important part that is investigated is the *Shift Register*. The gate counts and the area occupied by the *Shift Register* is given in Table 5.6.

Table 5.6: Gate types, counts and the area occupied by the Shift Register

Gate Type	Unit Area	Count	Total Area
AN2	2	1	2
AN2P	2	1	2
AO1	2	5	10
AO2	2	46	92
AO3	2	38	76
AO4	2	4	8
AO4P	4	1	4
AO6	2	2	4
AO7	2	11	22
AO7P	3	1	3
AO21	2	2	4
AO22P	3	2	6
AO211	3	1	3
AO222	4	4	16
BUF1	1	5	5
BUF4	2	1	2
EO1	3	5	15
EON1	3	1	3
FD1Q	6	29	174
IV	1	23	23
IVA	1	15	15
IVAP	2	2	4
IVP	1	16	16
ND2	1	30	30
ND2P	2	2	4

Table 5.6: Gate types, counts and the area occupied by the Shift Register

Gate Type	Unit Area	Count	Total Area
ND3	2	1	2
ND4	2	4	8
ND4P	4	1	4
NND2	2	2	4
NNR2	2	2	4
NR2	1	19	19
NR2P	2	3	6
NR3	2	2	4
NR4	2	4	8
OA21	2	25	50
OR2P	2	1	2
OR3	2	3	6
onepos_0	7	1	7
onepos_1	8	4	32
sorter	102	1	102
Net Interconnect	N/A	N/A	592.34
Total	**150**	**161**	**1393.34**
Max	**102**	**30**	**102**

The *Shift Register* occupies an area of 1393.34. Only one *Shift Register* is used in the design. The delay through the *Shift Register* is 5.82 ns.

The final modification is to the PC incrementer logic that is used in the IF stage for calculating the address of the next instruction to be fetched. The same block used in the in-order implementation is also used in the out-of-order implementation. The synthesized version of this block occupies an area of 329.42, and has a critical path delay of 3.77 ns. The synthesis output reports are given in Appendix 3.

5.3 Conclusions

In this chapter the hardware that is needed to be added to an out-of-order pipelined processor for support of *Guarded Execution* was investigated. In this section, the results obtained are analyzed.

When the data in Table 5.4 is compared with the data in Table 5.2, the results in Table 5.7 are obtained.

Table 5.7: Percent overhead of GUARDed Execution for each stage

% increase	IF	ID	IS	EX	MEM	Total
Combinational Area	54.58%	3.3%	26.25%	0%	0%	24.95%
Non-Combinational Area	-39.8%	N/A	1.60%	0%	0%	-2.43%
Net Interconnect Area	3.55%	3.36%	6.72%	0%	0%	6.00%
Total Cell Area	-12.38%	3.3&	20.40%	0%	0%	18.00%
Total Area	-8.16%	3.33%	16.24%	0%	0%	14.27%
Data Arrival Time	2.82%	0%	1.21%	0%	0%	1.21%

The most interesting part of Table 5.7 is that even though the bit count of the latches in the IF block does not change, the Non-Combinational Area of this block decreases. This is due to the latch type change between the two synthesis parts. Since delay is not constrained and the area is tried to be minimized, slower but more area efficient latches are used. The main impact on the minimum clock cycle of the pipeline comes from the IS stage since that stage determines the clock cycle.

The hardware added introduces a 14.27% area overhead to the out-of-order pipeline as opposed to the 14.72% overhead introduced to the in-order pipeline.

The cycle time of the processor is affected by 1.21%. This whole effect is due to the cycle time change of the IS stage.

Chapter 6

Conclusions

In this thesis, we have investigated the hardware cost of supporting *Guarded Execution* in an existing architecture. We have introduced three *Guard* type instructions which are proposed in [1] to a reduced DLX instruction set, and built an in-order and an out-of-order pipeline. Mostly the hardware organization suggested in [1] has been used.

After synthesizing the in-order pipelined design, it was clear that the cost of integrating *Guarded Execution* into an in-order pipelined architecture is justifiable. It does not have an effect on the cycle time for the pipeline studied, and the area overhead is approximately 15%.

After synthesizing the out-of-order pipelined design, we saw that the hardware cost of integrating *Guarded Execution* in terms of the additional area required went up when compared to the in-order case. The area overhead was approximately 14%. No cycle time overhead was found for the pipeline studied due to the long access times of the caches. Only the cycle time of the 'Instruction Execute' Stage was affected by 1.11%.

The cycle time of the in-order processor studied did not seem to be affected much because of the 64KB data and instruction caches, the access times to those caches pretty much determine the cycle time of the whole processor. But, if we consider the extreme case of having no caches at all, then the stage determining the cycle time of the processor will definitely be the EX stage, and the cycle time of the EX stage is affected only by 1.11% even though the synthesis tools were not set up to minimize the cycle time of the pipeline. For the out-of-order case, the long access times of the caches does not matter much, the IS stage is the stage that determines the cycle time. The cycle time of the processor went up by 1.21%. That is also justifiable since only area optimization is considered.

Thus we can conclude that the hardware overhead in terms of both the area and the clock cycle penalty of integrating *Guarded Execution* in an existing architecture is justifiable.

6.1 Future Work

There are some limitations [1] on the *Guarding Region* of the *GUARD* instructions like the one mentioned in section 4.3. An algorithm which could eliminate this kind of limitations can be come up with in the expense of complicating the hardware.

Also, integrating *Guarded Execution* into some more aggressive architectures should be studied since the state of the art processors are no longer simple pipelined machines.

REFERENCES

[1] Dionisios N. Pnevmatikatos, *Incorporating Guarded Execution into Existing Instruction Sets*. Ph.D. thesis, University of Wisconsin-Madison, 1996.

[2] Dionisios N. Pnevmatikatos and Gurindar S. Sohi. *Guarded Execution and Branch Prediction in Dynamic ILP Processors*. In *Proceedings of the 21st Annual International Symposium on Computer Architecture*, pages 120-129, Chicago, Illinois, April 18-21, 1994.

[3] Gerry Kane, *MIPS Risc Architecture*. Prentice Hall, 1988.

[4] MIPS Technologies Inc., *MIPS R10000 Microprocessor User's Manual Version 1.1*. 1995.

[5] John L. Hennessy and David A. Patterson, *Computer Architecture A Quantitative Approach*. Morgan Kaufmann Publishers Inc., 1995.

[6] John L. Hennessy and David A. Patterson, *Computer Architecture Hardware Software Interface*. Morgan Kaufmann Publishers Inc., 1995.

[7] James E. Smith, *A Study of Branch Prediction Strategies*. In *Proceedings of the 8th International Symposium on Computer Architecture*, pages 135-148, Minneapolis, Minnesota, May 12-14, 1981.

[8] MIPS Technologies Inc. *UMIPS-V Reference Manual*. Sunnyvale, California, 1990.

[9] Advanced RISC Machines, *ARM 610 RISC Processor*. January 1993.

[10] Peter Y. T. Hsu and Edward S. Davidson. *Highly Concurrent Scalar Processing*. In *Proceedings of the 13th Annual International Symposium on Computer Architecture*, pages 386-395, Tokyo, Japan, June 2-5, 1986.

Appendix 1

LCA500K Internal Macrocells Used

This appendix contains the definitions abbreviations for the gates used in this work. The suffix ‘p’ stands for a gate having a high output drive strength, otherwise gates having standard output drive strength are used. The rise and fall times are in nanoseconds and measured from 10% to 90% VDD.

Table A.1: Macrocell Abbreviations

Version	Name	Gate Count	Output	Standard Loads				
				2	4	8	12	16
an2	2-Input AND	2	tpLH	0.22	0.29	0.43	0.59	0.74
			tpHL	0.19	0.23	0.30	0.37	0.43
an2p	2-Input AND	2	tpLH	0.21	0.24	0.31	0.39	0.46
			tpHL	0.19	0.22	0.26	0.30	0.34
an3	3-Input AND	2	tpLH	0.28	0.35	0.50	0.66	0.81
			tpHL	0.22	0.26	0.34	0.40	0.47
an4	4-Input AND	3	tpLH	0.36	0.44	0.59	0.75	0.90
			tpHL	0.25	0.29	0.37	0.44	0.51
an4p	4-Input AND	3	tpLH	0.37	0.41	0.49	0.57	0.65
			tpHL	0.25	0.28	0.33	0.37	0.41
ao1	2-AND into 3-NOR	2	tpLH	0.43	0.66	1.11	1.55	1.99
			tpHL	0.13	0.18	0.27	0.36	0.45
ao2	2 2 ANDs into 2-NOR	2	tpLH	0.33	0.48	0.78	1.08	1.37
			tpHL	0.16	0.21	0.33	0.44	0.55

Table A.1: Macrocell Abbreviations

Version	Name	Gate Count	Output	Standard Loads				
				2	4	8	12	16
ao3	2-OR into 3-NAND	2	tpLH	0.22	0.33	0.55	0.78	1.01
			tpHL	0.20	0.28	0.43	0.59	0.75
ao4	2 2-ORs into 2-NAND	2	tpLH	0.30	0.45	0.76	1.05	1.34
			tpHL	0.17	0.23	0.34	0.45	0.56
ao6	2-AND into 2-NOR	2	tpLH	0.19	0.24	0.36	0.49	0.62
			tpHL	0.11	0.13	0.19	0.25	0.30
ao7	2-OR into 2-NAND	2	tpLH	0.22	0.34	0.59	0.84	1.09
			tpHL	0.15	0.20	0.31	0.42	0.54
ao7p	2-OR into 2-NAND	3	tpLH	0.16	0.22	0.34	0.46	0.59
			tpHL	0.12	0.15	0.20	0.26	0.31
ao11	3 2-ANDs into 3-NOR	5	tpLH	0.37	0.44	0.59	0.75	0.91
			tpHL	0.31	0.36	0.43	0.50	0.57
b1a	Inverting Power Buffer	3	tpLH	0.06	0.08	0.13	0.22	0.42
			tpHL	0.07	0.10	0.14	0.23	0.41
b4i	4 parallel IVs	2	tpLH	0.07	0.10	0.17	0.31	0.62
			tpHL	0.06	0.08	0.11	0.18	0.31
buf1	Buffer, Internal 1X Drive	1	tpLH	0.17	0.24	0.38	0.54	0.70
			tpHL	0.16	0.19	0.26	0.33	0.39
buf4	Buffer, Internal 3X Drive	2	tpLH	0.17	0.19	0.24	0.28	0.33
			tpHL	0.20	0.22	0.25	0.28	0.30
buf5	Buffer, Internal 4/2X Drive	3	tpLH	0.15	0.17	0.20	0.24	0.27
			tpHL	0.24	0.26	0.30	0.34	0.37
en	2-Input Exclusive NOR	3	tpLH	0.23	0.30	0.43	0.58	0.73
			tpHL	0.20	0.25	0.33	0.40	0.48

Table A.1: Macrocell Abbreviations

Version	Name	Gate Count	Output	Standard Loads				
				2	4	8	12	16
enp	2-Input Exclusive NOR	4	tpLH	0.22	0.26	0.33	0.39	0.46
			tpHL	0.20	0.23	0.29	0.34	0.38
eo	2-Input Exclusive OR	3	tpLH	0.23	0.30	0.42	0.57	0.73
			tpHL	0.19	0.24	0.33	0.40	0.47
eo1	2-AND, 2-NOR into 2-NOR	3	tpLH	0.30	0.45	0.75	1.05	1.34
			tpHL	0.15	0.21	0.32	0.43	0.54
fd1q	D Flip Flop	6	tpLH	0.57	0.64	0.79	0.95	1.10
			tpHL	0.58	0.63	0.70	0.77	0.84
fd1slq	D Flip Flop with Enable Scan	10	tpLH	0.53	0.60	0.75	0.90	1.06
			tpHL	0.51	0.57	0.66	0.74	0.82
fd2slq	D Flip Flop with Clear, Scan, Load	13	tpLH	0.62	0.69	0.84	1.00	1.15
			tpHL	0.59	0.63	0.70	0.77	0.84
ha	Half Adder	5	tpLH	0.45	0.52	0.67	0.82	0.98
			tpHL	0.46	0.51	0.59	0.67	0.74
iv	Inverter	1	tpLH	0.10	0.17	0.32	0.47	0.63
			tpHL	0.08	0.12	0.18	0.25	0.32
ivp	Inverter	1	tpLH	0.07	0.10	0.17	0.24	0.31
			tpHL	0.08	0.12	0.18	0.25	0.32
iva	Inverter with Parallel P Transistors	1	tpLH	0.07	0.11	0.17	0.25	0.32
			tpHL	0.09	0.12	0.18	0.25	0.32
ivap	Inverter with Parallel P Transistors	2	tpLH	0.06	0.07	0.11	0.14	0.17
			tpHL	0.07	0.09	0.12	0.15	0.18

Table A.1: Macrocell Abbreviations

Version	Name	Gate Count	Output	Standard Loads				
				2	4	8	12	16
mux211	Inverting Gate MUX	3	tpLH	0.12	0.15	0.22	0.29	0.36
			tpHL	0.14	0.17	0.24	0.31	0.38
nd2	2-Input NAND	1	tpLH	0.13	0.20	0.34	0.50	0.66
			tpHL	0.13	0.18	0.29	0.40	0.51
nd2p	2-Input NAND	2	tpLH	0.10	0.13	0.20	0.27	0.34
			tpHL	0.10	0.13	0.18	0.24	0.29
nd3	3-Input NAND	2	tpLH	0.15	0.22	0.37	0.52	0.68
			tpHL	0.17	0.25	0.41	0.56	0.72
nd3p	3-Input NAND	3	tpLH	0.12	0.15	0.22	0.29	0.36
			tpHL	0.13	0.17	0.25	0.33	0.41
nd4	4-Input NAND	2	tpLH	0.17	0.24	0.38	0.54	0.69
			tpHL	0.22	0.32	0.53	0.73	0.93
nd8	8-Input NAND	6	tpLH	0.32	0.39	0.54	0.70	0.85
			tpHL	0.46	0.50	0.58	0.65	0.73
nr2	2-Input NOR	1	tpLH	0.21	0.36	0.67	0.97	1.26
			tpHL	0.10	0.13	0.19	0.26	0.33
nr2p	2-Input NOR	2	tpLH	0.14	0.21	0.36	0.52	0.67
			tpHL	0.08	0.10	0.13	0.16	0.19
nr4	4-Input NOR	4	tpLH	0.53	0.84	1.43	2.02	2.61
			tpHL	0.11	0.15	0.22	0.28	0.35
or3	3-Input OR	2	tpLH	0.20	0.27	0.41	0.57	0.73
			tpHL	0.38	0.43	0.53	0.61	0.69
or4	4-Input OR	3	tpLH	0.21	0.28	0.42	0.58	0.74
			tpHL	0.50	0.56	0.66	0.76	0.85

Appendix 2

Synthesis Results

of the

In-Order Pipelined Architecture

The synthesis outputs of all the stages and the new hardware introduced are given. <stage_name>_before corresponds to the stage before the integration of GUARD type instructions while <stage_name>_after corresponds to the stage after the integration of GUARD type instructions.

IF_before

```
****************************************
Report : area
Design : fetch_stage
Version: v3.4a
Date   : Mon May  5 14:42:38 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:             253
Number of nets:              642
Number of cells:             420
Number of references:         17

Combinational area:          701.000000
Noncombinational area:       994.000000
Net Interconnect area:       671.742676

Total cell area:            1695.000000
Total area:                 2366.742676
1
design_analyzer>
****************************************
Report : reference
Design : fetch_stage
Version: v3.4a
Date   : Mon May  5 14:42:38 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO7                 lca500kv       2.000000       1      2.000000
B2A                 lca500kv       4.000000       1      4.000000
BUF4                lca500kv       2.000000       1      2.000000
BUF5                lca500kv       3.000000       2      6.000000
FD1                 lca500kv       7.000000       2     14.000000   n
```

```
FD1Q               lca500kv       6.000000      60    360.000000  n
FDS2LQ             lca500kv      10.000000      62    620.000000  n
IV                 lca500kv       1.000000       4      4.000000
IVA                lca500kv       1.000000      29     29.000000
IVAP               lca500kv       2.000000       2      4.000000
IVP                lca500kv       1.000000     124    124.000000
ND3                lca500kv       2.000000      30     60.000000
NND2               lca500kv       2.000000      10     20.000000
OA22               lca500kv       4.000000      58    232.000000
OA22P              lca500kv       4.000000       2      8.000000
OR2                lca500kv       2.000000      31     62.000000
incrementer                     144.000000       1    144.000000  h
---------------------------------------------------------------------------
Total 17 references                                     1695.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : fetch_stage
Version: v3.4a
Date   : Mon May  5 14:42:38 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
fetch_stage           B2X2               lca500kv
incrementer           B0.5X0.5           lca500kv
incrementer_DW01_inc_30_0
                      B0.5X0.5           lca500kv


  Startpoint: icache_hit (input port)
  Endpoint: if_pc_q_reg[0]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock (input port clock) (rise edge)     0.00       0.00
  input external delay                    13.80      13.80 f
  icache_hit (in)                          0.00      13.80 f
  U125/Z (IVP)                             0.05      13.85 r
  U349/Z (AO7)                             0.14      13.99 f
  U126/Z (NND2)                            0.25      14.24 r
  U135/Z (NND2)                            0.27      14.51 f
```

```
U25/Z (IV)                               0.18       14.69 r
U24/Z (IVAP)                             0.77       15.46 f
U343/Z (OA22)                            0.46       15.93 f
U122/Z (ND3)                             0.16       16.09 r
if_pc_q_reg[0]/D (FD1Q)                  0.00       16.09 r
data arrival time                                   16.09
-----------------------------------------------------------
(Path is unconstrained)
```

IF_after

```
****************************************
Report : area
Design : new_fetch_stage
Version: v3.4a
Date   : Mon May  5 15:04:15 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              258
Number of nets:               647
Number of cells:              420
Number of references:          17

Combinational area:        857.000000
Noncombinational area:     994.000000
Net Interconnect area:     703.696167

Total cell area:          1851.000000
Total area:               2554.696289
1
design_analyzer>
****************************************
Report : reference
Design : new_fetch_stage
Version: v3.4a
Date   : Mon May  5 15:04:15 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO7                 lca500kv       2.000000       1      2.000000
B2A                 lca500kv       4.000000       1      4.000000
BUF4                lca500kv       2.000000       1      2.000000
BUF5                lca500kv       3.000000       2      6.000000
FD1                 lca500kv       7.000000       2     14.000000   n
```

```
FD1Q              lca500kv      6.000000      60    360.000000  n
FDS2LQ            lca500kv     10.000000      62    620.000000  n
IV                lca500kv      1.000000      26     26.000000
IVA               lca500kv      1.000000       7      7.000000
IVAP              lca500kv      2.000000       2      4.000000
IVP               lca500kv      1.000000     124    124.000000
ND3               lca500kv      2.000000      30     60.000000
NND2              lca500kv      2.000000      10     20.000000
OA22              lca500kv      4.000000      58    232.000000
OA22P             lca500kv      4.000000       2      8.000000
OR2               lca500kv      2.000000      31     62.000000
incrementer                   300.000000       1    300.000000  h
-----------------------------------------------------------------------------
Total 17 references                                1851.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : new_fetch_stage
Version: v3.4a
Date   : Mon May  5 15:04:15 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
new_fetch_stage       B2X2              lca500kv
incrementer            B0.5X0.5          lca500kv
incrementer_DW01_inc_30_0
                       B0.5X0.5          lca500kv
incrementer_DW01_add_30_0
                       B0.5X0.5          lca500kv


  Startpoint: icache_hit (input port)
  Endpoint: if_pc_q_reg[0]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock (input port clock) (rise edge)     0.00       0.00
  input external delay                    13.80      13.80 f
  icache_hit (in)                          0.00      13.80 f
  U125/Z (IVP)                             0.05      13.85 r
  U349/Z (AO7)                             0.14      13.99 f
```

```
U126/Z (NND2)                              0.25       14.24 r
U135/Z (NND2)                              0.27       14.51 f
U25/Z (IV)                                 0.18       14.69 r
U24/Z (IVAP)                               0.77       15.46 f
U343/Z (OA22)                              0.46       15.93 f
U122/Z (ND3)                               0.16       16.09 r
if_pc_q_reg[0]/D (FD1Q)                    0.00       16.09 r
data arrival time                                     16.09
-----------------------------------------------------------
(Path is unconstrained)
```

ID_before

```
****************************************
Report : area
Design : dec_stage
Version: v3.4a
Date   : Mon May  5 17:08:22 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              451
Number of nets:               433
Number of cells:              209
Number of references:          23

Combinational area:        431.000000
Noncombinational area:    1004.000000
Net Interconnect area:     504.026978

Total cell area:          1435.000000
Total area:               1939.026978
1
design_analyzer>
****************************************
Report : reference
Design : dec_stage
Version: v3.4a
Date   : Mon May  5 17:08:22 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2P                lca500kv       2.000000       1      2.000000
AO7                 lca500kv       2.000000       1      2.000000
AO21N               lca500kv       2.000000       1      2.000000
BUF1                lca500kv       1.000000       1      1.000000
EN                  lca500kv       3.000000      15     45.000000
```

```
EO                 lca500kv       3.000000       1      3.000000
FD1Q               lca500kv       6.000000     155    930.000000  n
FD1S               lca500kv       9.000000       1      9.000000  n
FD3SQM             lca500kv      13.000000       5     65.000000  n
IV                 lca500kv       1.000000       4      4.000000
IVA                lca500kv       1.000000       2      2.000000
IVP                lca500kv       1.000000       5      5.000000
MUX21L             lca500kv       3.000000       1      3.000000
ND2                lca500kv       1.000000       3      3.000000
ND3                lca500kv       2.000000       2      4.000000
ND4                lca500kv       2.000000       1      2.000000
ND5                lca500kv       4.000000       2      8.000000
ND6                lca500kv       5.000000       3     15.000000
NND2               lca500kv       2.000000       1      2.000000
NR6                lca500kv       5.000000       1      5.000000
OA22               lca500kv       4.000000       1      4.000000
adder                           207.000000       1    207.000000  h
decoder                         112.000000       1    112.000000  h
-----------------------------------------------------------------------------
Total 23 references                                  1435.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : dec_stage
Version: v3.4a
Date   : Mon May  5 17:08:22 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
dec_stage            B1X1             lca500kv
decoder               B0.5X0.5          lca500kv
adder                 B0.5X0.5          lca500kv
adder_DW01_add_30_0    B0.5X0.5          lca500kv


  Startpoint: ex_A_d[0] (input port)
  Endpoint: ex_A_q_reg[0]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock (input port clock) (rise edge)     0.00       0.00
```

```
input external delay                        7.80       7.80 f
ex_A_d[0] (in)                              0.00       7.80 f
ex_A_q_reg[0]/D (FD1Q)                      0.00       7.80 f
data arrival time                                      7.80
-----------------------------------------------------------
(Path is unconstrained)
```

ID_after

```
****************************************
Report : area
Design : new_dec_stage
Version: v3.4a
Date   : Mon May  5 11:59:05 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              485
Number of nets:               494
Number of cells:              249
Number of references:          30

Combinational area:        547.000000
Noncombinational area:    1260.000000
Net Interconnect area:     752.618164

Total cell area:          1807.000000
Total area:               2559.618164
1
design_analyzer>
****************************************
Report : reference
Design : new_dec_stage
Version: v3.4a
Date   : Mon May  5 11:59:06 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2P                lca500kv       2.000000       1      2.000000
AO7                 lca500kv       2.000000       1      2.000000
AO21N               lca500kv       2.000000       1      2.000000
BUF1                lca500kv       1.000000       1      1.000000
BUF4                lca500kv       2.000000       1      2.000000
```

```
EN                  lca500kv       3.000000      15     45.000000
EO                  lca500kv       3.000000       2      6.000000
FD1Q                lca500kv       6.000000     167   1002.000000  n
FD1S                lca500kv       9.000000      12    108.000000  n
FD3SQM              lca500kv      13.000000       5     65.000000  n
FDS2LSQ             lca500kv      13.000000       1     13.000000  n
IV                  lca500kv       1.000000       4      4.000000
IVA                 lca500kv       1.000000       2      2.000000
IVP                 lca500kv       1.000000       6      6.000000
MUX21L              lca500kv       3.000000       1      3.000000
ND2                 lca500kv       1.000000       3      3.000000
ND3                 lca500kv       2.000000       2      4.000000
ND4                 lca500kv       2.000000       1      2.000000
ND5                 lca500kv       4.000000       2      8.000000
ND6                 lca500kv       5.000000       3     15.000000
NND2                lca500kv       2.000000       1      2.000000
NR6                 lca500kv       5.000000       1      5.000000
OA22                lca500kv       4.000000       1      4.000000
SFD2                lca500kv       8.000000       9     72.000000  n
adder                            207.000000       1    207.000000  h
decoder                          122.000000       1    122.000000  h
guarddec_0                        25.000000       1     25.000000  h
guarddec_1                        25.000000       1     25.000000  h
guarddec_2                        25.000000       1     25.000000  h
guarddec_3                        25.000000       1     25.000000  h
-----------------------------------------------------------------------------
Total 30 references                                   1807.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : new_dec_stage
Version: v3.4a
Date   : Mon May  5 11:59:06 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design            Wire Loading Model      Library
------------------------------------------------
new_dec_stage           B2X2               lca500kv
decoder                 B0.5X0.5           lca500kv
guarddec_3              B0.5X0.5           lca500kv
guarddec_2              B0.5X0.5           lca500kv
guarddec_1              B0.5X0.5           lca500kv
guarddec_0              B0.5X0.5           lca500kv
adder                   B0.5X0.5           lca500kv
adder_DW01_add_30_0     B0.5X0.5           lca500kv
```

```
Startpoint: id_ir_q[29]
            (input port)
Endpoint: ex_V_q_reg (rising edge-triggered flip-flop clocked by clock)
Path Group: clock
Path Type: max

Point                                      Incr       Path
-----------------------------------------------------------
clock (input port clock) (rise edge)       0.00       0.00
input external delay                       0.00       0.00 r
id_ir_q[29] (in)                           0.00       0.00 r
r3/instr[29] (decoder)                     0.00       0.00 r
r3/U166/Z (IVA)                            0.24       0.24 f
r3/U132/Z (NR3)                            0.48       0.72 r
r3/U159/Z (AO2)                            0.39       1.11 f
r3/U149/Z (AO6)                            0.39       1.50 r
r3/U150/Z (OA21)                           0.44       1.94 r
r3/U119/Z (NR3)                            0.25       2.19 f
r3/U173/Z (IVP)                            0.12       2.31 r
r3/U121/Z (NR2)                            0.22       2.53 f
r3/U147/Z (ND4)                            0.19       2.72 r
r3/U114/Z (AO3)                            0.27       2.99 f
r3/U99/Z (NR3)                             0.48       3.48 r
r3/U154/Z (ND4)                            0.43       3.91 f
r3/U109/Z (AO3)                            0.34       4.25 r
r3/decode_out[1] (decoder)                 0.00       4.25 r
U56/Z (IV)                                 0.20       4.45 f
U57/Z (ND4)                                0.19       4.64 r
U45/Z (BUF1)                               0.53       5.17 r
ex_rd_q_reg[3]/Q (FD3SQM)                  0.73       5.90 r
U90/Z (EN)                                 0.48       6.38 f
U59/Z (ND5)                                0.33       6.71 r
U64/Z (OA22)                               0.37       7.08 r
U100/Z (MUX21L)                            0.18       7.26 f
U83/Z (NND2)                               0.12       7.38 r
U58/Z (AO21N)                              0.18       7.80 f
ex_V_q_reg/TE (FDS2LSQ)                    0.00       7.80 f
data arrival time                                     7.80
-----------------------------------------------------------
(Path is unconstrained)
```

EX_before

```
****************************************
Report : area
Design : ex_stage
Version: v3.4a
Date   : Mon May  5 14:20:43 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              312
Number of nets:               577
Number of cells:              349
Number of references:          19

Combinational area:       2690.000000
Noncombinational area:     648.000000
Net Interconnect area:    2578.010986

Total cell area:          3338.000000
Total area:               5916.010742
1
design_analyzer>
****************************************
Report : reference
Design : ex_stage
Version: v3.4a
Date   : Mon May  5 14:20:43 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000       1      2.000000
AN3                 lca500kv       2.000000       2      4.000000
AO2                 lca500kv       2.000000      32     64.000000
AO3                 lca500kv       2.000000      32     64.000000
AO222               lca500kv       4.000000      64    256.000000
```

```
B4I              lca500kv       2.000000       1      2.000000
BUF4             lca500kv       2.000000       3      6.000000
BUF5             lca500kv       3.000000       4     12.000000
EN               lca500kv       3.000000      16     48.000000
EO               lca500kv       3.000000       4     12.000000
FD1Q             lca500kv       6.000000     108    648.000000  n
IVAP             lca500kv       2.000000       3      6.000000
IVP              lca500kv       1.000000      33     33.000000
ND2              lca500kv       1.000000      32     32.000000
ND4              lca500kv       2.000000       4      8.000000
NND2             lca500kv       2.000000       1      2.000000
NR2              lca500kv       1.000000       6      6.000000
NR4              lca500kv       2.000000       2      4.000000
alu                          2129.000000       1   2129.000000  h
----------------------------------------------------------------------------
Total 19 references                                3338.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : ex_stage
Version: v3.4a
Date   : Mon May  5 14:20:44 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
ex_stage               B2X2              lca500kv
alu                    B2X2              lca500kv
barrel_shifter         B1X1              lca500kv
adder_c                B1X1              lca500kv
add_31                 B0.5X0.5          lca500kv
add_30                 B0.5X0.5          lca500kv
add_29                 B0.5X0.5          lca500kv
add_28                 B0.5X0.5          lca500kv
add_27                 B0.5X0.5          lca500kv
cladder_9              B0.5X0.5          lca500kv
cladder_8              B0.5X0.5          lca500kv
add_26                 B0.5X0.5          lca500kv
add_25                 B0.5X0.5          lca500kv
add_24                 B0.5X0.5          lca500kv
cladder_7              B0.5X0.5          lca500kv
add_23                 B0.5X0.5          lca500kv
add_22                 B0.5X0.5          lca500kv
cladder_6              B0.5X0.5          lca500kv
add_21                 B0.5X0.5          lca500kv
```

```
add_20                 B0.5X0.5          lca500kv
add_19                 B0.5X0.5          lca500kv
add_18                 B0.5X0.5          lca500kv
add_17                 B0.5X0.5          lca500kv
cladder_5              B0.5X0.5          lca500kv
add_16                 B0.5X0.5          lca500kv
add_15                 B0.5X0.5          lca500kv
add_14                 B0.5X0.5          lca500kv
cladder_4              B0.5X0.5          lca500kv
add_13                 B0.5X0.5          lca500kv
add_12                 B0.5X0.5          lca500kv
add_11                 B0.5X0.5          lca500kv
add_10                 B0.5X0.5          lca500kv
cladder_3              B0.5X0.5          lca500kv
add_9                  B0.5X0.5          lca500kv
add_8                  B0.5X0.5          lca500kv
add_7                  B0.5X0.5          lca500kv
add_6                  B0.5X0.5          lca500kv
cladder_2              B0.5X0.5          lca500kv
add_5                  B0.5X0.5          lca500kv
add_4                  B0.5X0.5          lca500kv
add_3                  B0.5X0.5          lca500kv
add_2                  B0.5X0.5          lca500kv
cladder_1              B0.5X0.5          lca500kv
cladder1               B0.5X0.5          lca500kv
add_1                  B0.5X0.5          lca500kv
cladder_0              B0.5X0.5          lca500kv
add_0                  B0.5X0.5          lca500kv


  Startpoint: mem_rd_q_reg[0]
              (rising edge-triggered flip-flop)
  Endpoint: overflow (output port)
  Path Group: (none)
  Path Type: max

  Point                                     Incr       Path
  -----------------------------------------------------------
  mem_rd_q_reg[0]/CP (FD1Q)                 0.00       0.00 r
  mem_rd_q_reg[0]/Q (FD1Q)                  0.49       0.49 f
  U222/Z (EN)                               0.35       0.84 r
  U153/Z (AN3)                              0.29       1.13 r
  U143/Z (ND4)                              0.56       1.69 f
  U30/Z (B4I)                               0.71       2.40 r
  U154/Z (NR4)                              0.36       2.76 f
  U144/Z (ND4)                              0.33       3.09 r
  U41/Z (IVAP)                              0.83       3.93 f
  U33/Z (NR2)                               0.26       4.19 r
  U36/Z (BUF4)                              0.44       4.63 r
  U147/Z (NND2)                             0.20       4.83 f
  U42/Z (BUF5)                              0.78       5.61 f
  U77/Z (AO3)                               0.31       5.92 r
```

```
r4/B[1] (alu)                                 0.00        5.92 r
r4/U358/Z (BUF4)                              0.49        6.41 r
r4/U309/Z (IVP)                               0.15        6.56 f
r4/U115/Z (AO2)                               0.29        6.85 r
r4/u2/operand2[1] (adder_c)                   0.00        6.85 r
r4/u2/U107/Z (BUF1)                           0.41        7.26 r
r4/u2/U73/Z (EO1)                             0.80        8.06 r
r4/u2/U39/Z (NNR2)                            0.36        8.42 f
r4/u2/r11/g[1] (cladder_7)                    0.00        8.42 f
r4/u2/r11/U11/Z (AO21)                        0.28        8.70 f
r4/u2/r11/U12/Z (AO21)                        0.30        9.00 f
r4/u2/r11/U8/Z (AO21)                         0.43        9.43 f
r4/u2/r11/go (cladder_7)                      0.00        9.43 f
r4/u2/r3/g[0] (cladder_8)                     0.00        9.43 f
r4/u2/r3/U11/Z (AO21)                         0.30        9.73 f
r4/u2/r3/U12/Z (AO21)                         0.30       10.04 f
r4/u2/r3/U8/Z (AO21)                          0.38       10.42 f
r4/u2/r3/go (cladder_8)                       0.00       10.42 f
r4/u2/r1/g[0] (cladder1)                      0.00       10.42 f
r4/u2/r1/U3/Z (AO21)                          0.38       10.80 f
r4/u2/r1/cout[0] (cladder1)                   0.00       10.80 f
r4/u2/r2/cin (cladder_5)                      0.00       10.80 f
r4/u2/r2/U6/Z (AO21)                          0.40       11.20 f
r4/u2/r2/U7/Z (AO21)                          0.48       11.68 f
r4/u2/r2/U9/Z (AO21)                          0.36       12.04 f
r4/u2/r2/cout[2] (cladder_5)                  0.00       12.04 f
r4/u2/r4/cin (cladder_9)                      0.00       12.04 f
r4/u2/r4/U6/Z (AO21)                          0.36       12.40 f
r4/u2/r4/U7/Z (AO21)                          0.43       12.83 f
r4/u2/r4/U9/Z (AO21)                          0.36       13.19 f
r4/u2/r4/cout[2] (cladder_9)                  0.00       13.19 f
r4/u2/U42/Z (EO)                              0.31       13.50 r
r4/u2/overflow (adder_c)                      0.00       13.50 r
r4/overflow (alu)                             0.00       13.50 r
overflow (out)                                0.00       13.50 r
data arrival time                                        13.50
-----------------------------------------------------------
(Path is unconstrained)
```

EX_after

```
****************************************
Report : area
Design : new_ex_stage
Version: v3.4a
Date   : Mon May  5 12:46:27 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              372
Number of nets:               694
Number of cells:              407
Number of references:          25

Combinational area:       2995.000000
Noncombinational area:     858.000000
Net Interconnect area:    2916.279785

Total cell area:          3853.000000
Total area:               6769.279785
1
design_analyzer>
****************************************
Report : reference
Design : new_ex_stage
Version: v3.4a
Date   : Mon May  5 12:46:27 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv      2.000000       2      4.000000
AN3                 lca500kv      2.000000       2      4.000000
AN4                 lca500kv      3.000000       1      3.000000
AN4P                lca500kv      3.000000       1      3.000000
AO2                 lca500kv      2.000000      44     88.000000
```

```
AO3                 lca500kv       2.000000      32     64.000000
AO222               lca500kv       4.000000      64    256.000000
B4I                 lca500kv       2.000000       1      2.000000
BUF4                lca500kv       2.000000       5     10.000000
BUF5                lca500kv       3.000000       4     12.000000
EN                  lca500kv       3.000000      16     48.000000
EO                  lca500kv       3.000000       5     15.000000
FD1Q                lca500kv       6.000000     108    648.000000  n
IV                  lca500kv       1.000000       1      1.000000
IVAP                lca500kv       2.000000       3      6.000000
IVP                 lca500kv       1.000000      46     46.000000
ND2                 lca500kv       1.000000      44     44.000000
ND2P                lca500kv       2.000000       1      2.000000
ND4                 lca500kv       2.000000       5     10.000000
NND2                lca500kv       2.000000       1      2.000000
NNR2                lca500kv       2.000000       1      2.000000
NR2                 lca500kv       1.000000       8      8.000000
NR4                 lca500kv       2.000000      10     20.000000
alu                             1940.000000       1   1940.000000  h
shiftreg                         615.000000       1    615.000000  h, n
-----------------------------------------------------------------------------
Total 25 references                                   3853.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : new_ex_stage
Version: v3.4a
Date   : Mon May  5 12:46:28 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
new_ex_stage           B2X2              lca500kv
alu                    B2X2              lca500kv
barrel_shifter         B1X1              lca500kv
adder_c                B1X1              lca500kv
add_31                 B0.5X0.5          lca500kv
add_30                 B0.5X0.5          lca500kv
add_29                 B0.5X0.5          lca500kv
add_28                 B0.5X0.5          lca500kv
add_27                 B0.5X0.5          lca500kv
cladder_9              B0.5X0.5          lca500kv
cladder_8              B0.5X0.5          lca500kv
add_26                 B0.5X0.5          lca500kv
add_25                 B0.5X0.5          lca500kv
```

```
add_24                 B0.5X0.5          lca500kv
cladder_7              B0.5X0.5          lca500kv
add_23                 B0.5X0.5          lca500kv
add_22                 B0.5X0.5          lca500kv
cladder_6              B0.5X0.5          lca500kv
add_21                 B0.5X0.5          lca500kv
add_20                 B0.5X0.5          lca500kv
add_19                 B0.5X0.5          lca500kv
add_18                 B0.5X0.5          lca500kv
add_17                 B0.5X0.5          lca500kv
cladder_5              B0.5X0.5          lca500kv
add_16                 B0.5X0.5          lca500kv
add_15                 B0.5X0.5          lca500kv
add_14                 B0.5X0.5          lca500kv
cladder_4              B0.5X0.5          lca500kv
add_13                 B0.5X0.5          lca500kv
add_12                 B0.5X0.5          lca500kv
add_11                 B0.5X0.5          lca500kv
add_10                 B0.5X0.5          lca500kv
cladder_3              B0.5X0.5          lca500kv
add_9                  B0.5X0.5          lca500kv
add_8                  B0.5X0.5          lca500kv
add_7                  B0.5X0.5          lca500kv
add_6                  B0.5X0.5          lca500kv
cladder_2              B0.5X0.5          lca500kv
add_5                  B0.5X0.5          lca500kv
add_4                  B0.5X0.5          lca500kv
add_3                  B0.5X0.5          lca500kv
add_2                  B0.5X0.5          lca500kv
cladder_1              B0.5X0.5          lca500kv
cladder1               B0.5X0.5          lca500kv
add_1                  B0.5X0.5          lca500kv
cladder_0              B0.5X0.5          lca500kv
add_0                  B0.5X0.5          lca500kv
shiftreg               B1X1              lca500kv
onepos_4               B0.5X0.5          lca500kv
onepos_3               B0.5X0.5          lca500kv
onepos_2               B0.5X0.5          lca500kv
onepos_1               B0.5X0.5          lca500kv
onepos_0               B0.5X0.5          lca500kv


  Startpoint: mem_rd_q_reg[0]
              (rising edge-triggered flip-flop)
  Endpoint: overflow (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  mem_rd_q_reg[0]/CP (FD1Q)                0.00       0.00 r
  mem_rd_q_reg[0]/Q (FD1Q)                 0.49       0.49 f
```

```
U263/Z (EN)                                 0.35       0.84 r
U182/Z (AN3)                                0.29       1.13 r
U171/Z (ND4)                                0.56       1.69 f
U59/Z (B4I)                                 0.70       2.39 r
U183/Z (NR4)                                0.34       2.73 f
U170/Z (ND4)                                0.33       3.06 r
U54/Z (IVAP)                                0.83       3.90 f
U42/Z (NR2)                                 0.25       4.15 r
U45/Z (BUF4)                                0.44       4.59 r
U176/Z (NND2)                               0.20       4.79 f
U55/Z (BUF5)                                0.78       5.57 f
U119/Z (AO3)                                0.31       5.88 r
r4/B[5] (alu)                               0.00       5.88 r
r4/U336/Z (BUF1)                            0.67       6.55 r
r4/U305/Z (IVP)                             0.20       6.75 f
r4/U99/Z (AO2)                              0.30       7.05 r
r4/u2/operand2[5] (adder_c)                 0.00       7.05 r
r4/u2/U102/Z (BUF1)                         0.41       7.46 r
r4/u2/U57/Z (EO1)                           0.80       8.26 r
r4/u2/U35/Z (NNR2)                          0.36       8.62 f
r4/u2/r10/g[1] (cladder_6)                  0.00       8.62 f
r4/u2/r10/U11/Z (AO21)                      0.28       8.90 f
r4/u2/r10/U12/Z (AO21)                      0.30       9.20 f
r4/u2/r10/U8/Z (AO21)                       0.43       9.63 f
r4/u2/r10/go (cladder_6)                    0.00       9.63 f
r4/u2/r3/g[1] (cladder_8)                   0.00       9.63 f
r4/u2/r3/U11/Z (AO21)                       0.25       9.88 f
r4/u2/r3/U12/Z (AO21)                       0.30      10.19 f
r4/u2/r3/U8/Z (AO21)                        0.38      10.57 f
r4/u2/r3/go (cladder_8)                     0.00      10.57 f
r4/u2/r1/g[0] (cladder1)                    0.00      10.57 f
r4/u2/r1/U3/Z (AO21)                        0.38      10.95 f
r4/u2/r1/cout[0] (cladder1)                 0.00      10.95 f
r4/u2/r2/cin (cladder_5)                    0.00      10.95 f
r4/u2/r2/U6/Z (AO21)                        0.40      11.35 f
r4/u2/r2/U7/Z (AO21)                        0.48      11.83 f
r4/u2/r2/U9/Z (AO21)                        0.36      12.19 f
r4/u2/r2/cout[2] (cladder_5)                0.00      12.19 f
r4/u2/r4/cin (cladder_9)                    0.00      12.19 f
r4/u2/r4/U6/Z (AO21)                        0.36      12.55 f
r4/u2/r4/U7/Z (AO21)                        0.43      12.98 f
r4/u2/r4/U9/Z (AO21)                        0.36      13.34 f
r4/u2/r4/cout[2] (cladder_9)                0.00      13.34 f
r4/u2/U42/Z (EO)                            0.31      13.65 r
r4/u2/overflow (adder_c)                    0.00      13.65 r
r4/overflow (alu)                           0.00      13.65 r
overflow (out)                              0.00      13.65 r
data arrival time                                     13.65
-----------------------------------------------------------
(Path is unconstrained)
```

MEM

```
****************************************
Report : area
Design : mem_stage
Version: v3.4a
Date   : Sat May  3 17:56:16 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              242
Number of nets:               215
Number of cells:              107
Number of references:           2

Combinational area:          2.000000
Noncombinational area:     636.000000
Net Interconnect area:     172.583725

Total cell area:           638.000000
Total area:                810.583740
1
design_analyzer>
****************************************
Report : reference
Design : mem_stage
Version: v3.4a
Date   : Sat May  3 17:56:21 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000       1      2.000000
FD1Q                lca500kv       6.000000     106    636.000000  n
-----------------------------------------------------------------------------
Total 2 references                                     638.000000
1
```

```
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : mem_stage
Version: v3.4a
Date   : Sat May  3 17:56:26 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
mem_stage             B1X1              lca500kv


  Startpoint: wb_loaddata_d[0]
              (input port)
  Endpoint: wb_loaddata_q_reg[0]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock (input port clock) (rise edge)     0.00       0.00
  input external delay                    13.80      13.80 f
  wb_loaddata_d[0] (in)                    0.00      13.80 f
  wb_loaddata_q_reg[0]/D (FD1Q)            0.00      13.80 f
  data arrival time                                  13.80
  -----------------------------------------------------------
  (Path is unconstrained)
```

WB

```
****************************************
Report : area
Design : wb_stage
Version: v3.4a
Date   : Sat May  3 18:21:43 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              133
Number of nets:               138
Number of cells:               38
Number of references:           7

Combinational area:        139.000000
Noncombinational area:       0.000000
Net Interconnect area:     115.248398

Total cell area:           139.000000
Total area:                254.248398
1
design_analyzer>
****************************************
Report : reference
Design : wb_stage
Version: v3.4a
Date   : Sat May  3 18:21:48 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000       1      2.000000
AO21                lca500kv       2.000000       1      2.000000
AO222               lca500kv       4.000000      32    128.000000
BUF5                lca500kv       3.000000       1      3.000000
IVAP                lca500kv       2.000000       1      2.000000
```

```
IVP                 lca500kv       1.000000       1      1.000000
NR2                 lca500kv       1.000000       1      1.000000
---------------------------------------------------------------------------
Total 7 references                                     139.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : wb_stage
Version: v3.4a
Date   : Sat May  3 18:21:54 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
wb_stage              B0.5X0.5          lca500kv


  Startpoint: wb_decode_q[0]
              (input port)
  Endpoint: gpr_writedata[31]
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                     0.00       0.00 f
  wb_decode_q[0] (in)                      0.00       0.00 f
  U12/Z (AO21)                             0.43       0.43 f
  U11/Z (IVAP)                             0.50       0.93 r
  U46/Z (NR2)                              0.25       1.18 f
  U10/Z (BUF5)                             0.81       1.99 f
  U13/Z (AO222)                            0.54       2.53 f
  gpr_writedata[31] (out)                  0.00       2.53 f
  data arrival time                                   2.53
  -----------------------------------------------------------
  (Path is unconstrained)
```

The Modified PC Incrementer Logic

```
****************************************
Report : area
Design : incrementer
Version: v3.4a
Date   : Sun Apr 20 23:32:37 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:               65
Number of nets:               115
Number of cells:               46
Number of references:          18

Combinational area:        223.000000
Noncombinational area:       0.000000
Net Interconnect area:     106.420853

Total cell area:           223.000000
Total area:                329.420837
1
design_analyzer>
****************************************
Report : reference
Design : incrementer
Version: v3.4a
Date   : Sun Apr 20 23:32:38 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
EO                  lca500kv       3.000000      25     75.000000
HA1                 lca500kv       5.000000       5     25.000000  r
add1                               3.000000       1      3.000000  h
add_0                              6.000000       1      6.000000  h
add_1                              6.000000       1      6.000000  h
```

```
add_2                            6.000000       1      6.000000  h
add_3                            6.000000       1      6.000000  h
cladder2                        12.000000       1     12.000000  h
cladder3_0                       9.000000       1      9.000000  h
cladder3_1                       9.000000       1      9.000000  h
cladder3_2                       9.000000       1      9.000000  h
cladder3_3                       9.000000       1      9.000000  h
cladder3_4                       9.000000       1      9.000000  h
cladder3_5                       6.000000       1      6.000000  h
cladder4                         2.000000       1      2.000000  h
cladder5                        14.000000       1     14.000000  h
cladder6                        15.000000       1     15.000000  h
cladder7                         2.000000       1      2.000000  h
-----------------------------------------------------------------------------
Total 18 references                                  223.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : incrementer
Version: v3.4a
Date   : Sun Apr 20 23:32:38 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
incrementer            B0.5X0.5          lca500kv
add1                   B0.5X0.5          lca500kv
add_3                  B0.5X0.5          lca500kv
add_2                  B0.5X0.5          lca500kv
add_1                  B0.5X0.5          lca500kv
add_0                  B0.5X0.5          lca500kv
cladder6               B0.5X0.5          lca500kv
cladder4               B0.5X0.5          lca500kv
cladder5               B0.5X0.5          lca500kv
cladder2               B0.5X0.5          lca500kv
cladder3_5             B0.5X0.5          lca500kv
cladder3_4             B0.5X0.5          lca500kv
cladder3_3             B0.5X0.5          lca500kv
cladder3_2             B0.5X0.5          lca500kv
cladder3_1             B0.5X0.5          lca500kv
cladder7               B0.5X0.5          lca500kv
cladder3_0             B0.5X0.5          lca500kv


  Startpoint: in1[1] (input port)
```

```
Endpoint: out[27] (output port)
Path Group: (none)
Path Type: max

Point                                     Incr       Path
-----------------------------------------------------------
input external delay                      0.00       0.00 r
in1[1] (in)                               0.00       0.00 r
U7/S (HA1)                                0.41       0.41 f
r11/p[1] (cladder6)                       0.00       0.41 f
r11/U11/Z (AO21)                          0.38       0.79 f
r11/U12/Z (AO21)                          0.30       1.09 f
r11/U8/Z (AO21)                           0.42       1.51 f
r11/go (cladder6)                         0.00       1.51 f
r3/g[0] (cladder5)                        0.00       1.51 f
r3/U11/Z (AN2)                            0.19       1.70 f
r3/U8/Z (AO3)                             0.22       1.92 r
r3/U12/Z (IV)                             0.13       2.05 f
r3/go (cladder5)                          0.00       2.05 f
r1/g (cladder7)                           0.00       2.05 f
r1/U2/Z (OR2)                             0.33       2.38 f
r1/cout (cladder7)                        0.00       2.38 f
r2/cin (cladder3_5)                       0.00       2.38 f
r2/U7/Z (AN2P)                            0.24       2.62 f
r2/U8/Z (AN2P)                            0.23       2.85 f
r2/cout[1] (cladder3_5)                   0.00       2.85 f
r5/cin (cladder3_3)                       0.00       2.85 f
r5/U7/Z (AN2P)                            0.21       3.07 f
r5/U8/Z (AN2P)                            0.21       3.28 f
r5/U5/Z (AN2)                             0.20       3.48 f
r5/cout[2] (cladder3_3)                   0.00       3.48 f
U16/Z (EO)                                0.29       3.77 r
out[27] (out)                             0.00       3.77 r
data arrival time                                    3.77
-----------------------------------------------------------
(Path is unconstrained)
```

The Guard Mask Decoder for GUARDBOTH Instructions

```
****************************************
Report : area
Design : guarddec
Version: v3.4a
Date   : Mon Apr 14 00:42:03 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:               11
Number of nets:                19
Number of cells:               14
Number of references:          11

Combinational area:         25.000000
Noncombinational area:      36.000000
Net Interconnect area:      20.107166

Total cell area:            61.000000
Total area:                 81.107166
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : guarddec
Version: v3.4a
Date   : Mon Apr 14 00:42:03 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
guarddec              B0.5X0.5          lca500kv


  Startpoint: in[0] (input port)
  Endpoint: tf[2] (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
```

```
input external delay                          0.00       0.00 f
in[0] (in)                                    0.00       0.00 f
U26/Z (IVP)                                   0.07       0.07 r
U20/Z (AO21N)                                 0.29       0.36 r
U19/Z (ND2)                                   0.17       0.53 f
U25/Z (AO2)                                   0.23       0.76 r
U15/Z (AO7)                                   0.26       1.02 f
U21/Z (NNR2)                                  0.19       1.21 f
U17/Z (AO6)                                   0.11       1.32 r
tf[2] (out)                                   0.00       1.32 r
data arrival time                                        1.32
-----------------------------------------------------------
(Path is unconstrained)
```

The Hardware Added to The EX Stage

```
****************************************
Report : area
Design : addition
Version: v3.4a
Date   : Sun Apr 20 20:28:52 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              131
Number of nets:               160
Number of cells:               77
Number of references:          15

Combinational area:        145.000000
Noncombinational area:       0.000000
Net Interconnect area:     122.604675

Total cell area:           145.000000
Total area:                267.604675
1
design_analyzer>
****************************************
Report : reference
Design : addition
Version: v3.4a
Date   : Sun Apr 20 20:28:52 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000      10     20.000000
AO2                 lca500kv       2.000000      12     24.000000
AO22P               lca500kv       3.000000      12     36.000000
BUF4                lca500kv       2.000000       2      4.000000
ENP                 lca500kv       4.000000       1      4.000000
```

```
EO                  lca500kv      3.000000       1      3.000000
IV                  lca500kv      1.000000       1      1.000000
IVA                 lca500kv      1.000000       1      1.000000
IVP                 lca500kv      1.000000      13     13.000000
ND2                 lca500kv      1.000000      12     12.000000
ND2P                lca500kv      2.000000       1      2.000000
ND8                 lca500kv      6.000000       1      6.000000
NNR2                lca500kv      2.000000       1      2.000000
NR2                 lca500kv      1.000000       1      1.000000
NR4                 lca500kv      2.000000       8     16.000000
-----------------------------------------------------------------------------
Total 15 references                                   145.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : addition
Version: v3.4a
Date   : Sun Apr 20 20:28:52 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
addition              B0.5X0.5          lca500kv


  Startpoint: alu_A[28] (input port)
  Endpoint: cond[11] (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                     0.00       0.00 f
  alu_A[28] (in)                           0.00       0.00 f
  U61/Z (NR4)                              0.54       0.54 r
  U71/Z (ND8)                              0.63       1.17 f
  U69/Z (NNR2)                             0.19       1.36 f
  U26/Z (BUF4)                             0.36       1.72 f
  U70/Z (NR2)                              0.18       1.90 r
  U25/Z (BUF4)                             0.39       2.29 r
  U82/Z (AO2)                              0.23       2.52 f
  U28/Z (ND2)                              0.09       2.61 r
  cond[11] (out)                           0.00       2.61 r
  data arrival time                                   2.61
  -----------------------------------------------------------
```

(Path is unconstrained)

One's Position Detector Circuit

```
****************************************
Report : area
Design : onepos
Version: v3.4a
Date   : Mon Apr 14 02:13:08 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:                8
Number of nets:                 9
Number of cells:                5
Number of references:           4

Combinational area:          8.000000
Noncombinational area:       0.000000
Net Interconnect area:       7.356280

Total cell area:             8.000000
Total area:                 15.356280
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : onepos
Version: v3.4a
Date   : Mon Apr 14 02:13:08 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
onepos                B0.5X0.5          lca500kv


  Startpoint: ins[2] (input port)
  Endpoint: outs[3] (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
```

```
input external delay                      0.00       0.00 f
ins[2] (in)                               0.00       0.00 f
U9/Z (NR4)                                0.41       0.41 r
outs[3] (out)                             0.00       0.41 r
data arrival time                                    0.41
-----------------------------------------------------------
(Path is unconstrained)
```

The Shift Register Design

```
****************************************
Report : area
Design : shiftreg
Version: v3.4a
Date   : Sun Apr 20 22:54:42 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              73
Number of nets:              337
Number of cells:             276
Number of references:         35

Combinational area:        436.000000
Noncombinational area:     126.000000
Net Interconnect area:     412.438477

Total cell area:           562.000000
Total area:                974.438477
1
design_analyzer>
****************************************
Report : reference
Design : shiftreg
Version: v3.4a
Date   : Sun Apr 20 22:54:42 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2P                lca500kv       2.000000       2      4.000000
AN3                 lca500kv       2.000000       2      4.000000
AO1                 lca500kv       2.000000       5     10.000000
AO2                 lca500kv       2.000000      29     58.000000
AO3                 lca500kv       2.000000      36     72.000000
```

```
AO4                 lca500kv       2.000000       2      4.000000
AO6                 lca500kv       2.000000       5     10.000000
AO7                 lca500kv       2.000000       1      2.000000
AO7P                lca500kv       3.000000       1      3.000000
AO11                lca500kv       5.000000       1      5.000000
AO21                lca500kv       2.000000       1      2.000000
AO21N               lca500kv       2.000000      17     34.000000
AO22P               lca500kv       3.000000       1      3.000000
AO211               lca500kv       3.000000       1      3.000000
AO222               lca500kv       4.000000       1      4.000000
BUF1                lca500kv       1.000000       2      2.000000
BUF4                lca500kv       2.000000       4      8.000000
EO1                 lca500kv       3.000000       2      6.000000
FD1Q                lca500kv       6.000000      21    126.000000  n
IV                  lca500kv       1.000000      36     36.000000
IVA                 lca500kv       1.000000      22     22.000000
IVP                 lca500kv       1.000000       4      4.000000
ND2                 lca500kv       1.000000      27     27.000000
ND2P                lca500kv       2.000000       2      4.000000
ND3                 lca500kv       2.000000       2      4.000000
ND3P                lca500kv       3.000000       1      3.000000
ND4                 lca500kv       2.000000       2      4.000000
NNR2                lca500kv       2.000000       2      4.000000
NR2                 lca500kv       1.000000      29     29.000000
NR2P                lca500kv       2.000000       1      2.000000
NR4                 lca500kv       2.000000       4      8.000000
OA222               lca500kv       4.000000       2      8.000000
OR3                 lca500kv       2.000000       2      4.000000
OR4                 lca500kv       3.000000       1      3.000000
onepos                             8.000000       5     40.000000  h
-----------------------------------------------------------------------------
Total 35 references                                    562.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : shiftreg
Version: v3.4a
Date   : Sun Apr 20 22:54:43 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
shiftreg               B1X1              lca500kv
onepos                 B0.5X0.5          lca500kv
```

```
Startpoint: write (input port)
Endpoint: count[1] (output port)
Path Group: (none)
Path Type: max

Point                                    Incr       Path
-----------------------------------------------------------
input external delay                     0.00       0.00 r
write (in)                               0.00       0.00 r
U265/Z (IVA)                             0.65       0.65 f
U159/Z (NR2)                             0.25       0.90 r
U145/Z (AO21N)                           0.40       1.30 r
U291/Z (IV)                              0.23       1.53 f
U119/Z (NR4)                             0.52       2.05 r
U75/Z (BUF1)                             0.38       2.43 r
U269/Z (IV)                              0.23       2.66 f
U176/Z (NR2)                             0.19       2.85 r
U120/Z (ND4)                             0.49       3.34 f
U121/Z (NR4)                             0.36       3.70 r
U76/Z (BUF1)                             0.43       4.14 r
U89/Z (ND4)                              0.27       4.41 f
U77/Z (IV)                               0.38       4.79 r
U180/Z (AO22P)                           0.43       5.22 r
U122/Z (AO1)                             0.20       5.42 f
U79/Z (ND3P)                             0.38       5.80 r
count[1] (out)                           0.00       5.80 r
data arrival time                                   5.80
-----------------------------------------------------------
(Path is unconstrained)
```

Appendix 3

Synthesis Results

of the

Out-Of-Order Pipelined Architecture

The synthesis outputs of all the stages and the new hardware introduced are given. <stage_name>_before corresponds to the stage before the integration of GUARD type instructions while <stage_name>_after corresponds to the stage after the integration of GUARD type instructions.

IF_before

```
****************************************
Report : area
Design : pipeline
Version: v3.4a
Date   : Mon May 12 11:17:40 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              221
Number of nets:               687
Number of cells:              408
Number of references:          15

Combinational area:        502.000000
Noncombinational area:    1226.000000
Net Interconnect area:     623.703430

Total cell area:          1728.000000
Total area:               2351.703369
1
design_analyzer>
****************************************
Report : reference
Design : pipeline
Version: v3.4a
Date   : Mon May 12 11:17:40 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO6                 lca500kv      2.000000       1      2.000000
B1A                 lca500kv      3.000000       3      9.000000
BUF5                lca500kv      3.000000       1      3.000000
FD1Q                lca500kv      6.000000       1      6.000000  n
FD1SLQ              lca500kv     10.000000      30    300.000000  n
```

```
FDS2LQ            lca500kv      10.000000      92    920.000000  n
IVA               lca500kv       1.000000      29     29.000000
IVAP              lca500kv       2.000000       1      2.000000
IVP               lca500kv       1.000000     122    122.000000
MUX21L            lca500kv       3.000000      30     90.000000
ND2               lca500kv       1.000000       2      2.000000
ND2P              lca500kv       2.000000       3      6.000000
NND2              lca500kv       2.000000       1      2.000000
NR2               lca500kv       1.000000      91     91.000000
incrementer                    144.000000       1    144.000000  h
---------------------------------------------------------------------------
Total 15 references                                 1728.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pipeline
Version: v3.4a
Date   : Mon May 12 11:17:41 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
pipeline               B2X2              lca500kv
incrementer            B0.5X0.5          lca500kv
incrementer_DW01_inc_30_0
                       B0.5X0.5          lca500kv


  Startpoint: btb_hit (input port)
  Endpoint: if_pc_q_reg[0]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock (input port clock) (rise edge)     0.00       0.00
  input external delay                    12.70      12.70 f
  btb_hit (in)                             0.00      12.70 f
  U40/Z (ND2)                              0.20      12.90 r
  U39/Z (B1A)                              0.71      13.61 f
  U254/Z (MUX21L)                          0.38      13.99 f
  U435/Z (IVP)                             0.06      14.05 r
  U375/Z (NR2)                             0.10      14.15 f
  U465/Z (NR2)                             0.26      14.42 r
```

```
  U130/Z (IVP)                              0.12      14.54 f
  if_pc_q_reg[0]/TE (FD1SLQ)                0.00      14.54 f
  data arrival time                                   14.54
  -----------------------------------------------------------
  (Path is unconstrained)
1
design_analyzer>
```

IF_after

```
****************************************
Report : area
Design : pipeline
Version: v3.4a
Date   : Tue May 13 00:13:24 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              226
Number of nets:               479
Number of cells:              347
Number of references:          14

Combinational area:        776.000000
Noncombinational area:     738.000000
Net Interconnect area:     645.855896

Total cell area:          1514.000000
Total area:               2159.855957
1
design_analyzer>
****************************************
Report : reference
Design : pipeline
Version: v3.4a
Date   : Tue May 13 00:13:24 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO2                 lca500kv       2.000000      30     60.000000
AO3                 lca500kv       2.000000      30     60.000000
AO6                 lca500kv       2.000000       1      2.000000
AO21N               lca500kv       2.000000       1      2.000000
AO22P               lca500kv       3.000000      92    276.000000
```

```
B1A                lca500kv       3.000000       2      6.000000
BUF4               lca500kv       2.000000       2      4.000000
FD1Q               lca500kv       6.000000     123    738.000000  n
IV                 lca500kv       1.000000      29     29.000000
IVAP               lca500kv       2.000000       1      2.000000
IVP                lca500kv       1.000000       1      1.000000
ND2                lca500kv       1.000000      31     31.000000
NR2                lca500kv       1.000000       3      3.000000
incrementer                     300.000000       1    300.000000  h
-----------------------------------------------------------------------------
Total 14 references                                  1514.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pipeline
Version: v3.4a
Date   : Tue May 13 00:13:25 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
pipeline               B1X1              lca500kv
incrementer            B0.5X0.5          lca500kv
incrementer_DW01_inc_30_0
                       B0.5X0.5          lca500kv
incrementer_DW01_add_30_0
                       B0.5X0.5          lca500kv


  Startpoint: btb_hit (input port)
  Endpoint: if_pc_q_reg[0]
            (rising edge-triggered flip-flop clocked by clock)
  Path Group: clock
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock (input port clock) (rise edge)     0.00       0.00
  input external delay                    12.70      12.70 r
  btb_hit (in)                             0.00      12.70 r
  U71/Z (AO21N)                            0.22      12.92 r
  U34/Z (BUF4)                             0.70      13.62 r
  U33/Z (ND2)                              0.39      14.01 f
  U32/Z (IVAP)                             0.51      14.52 r
  U224/Z (ND2)                             0.25      14.77 f
```

```
  U68/Z (AO3)                              0.18      14.95 r
  if_pc_q_reg[0]/D (FD1Q)                  0.00      14.95 r
  data arrival time                                  14.95
  -----------------------------------------------------------
  (Path is unconstrained)

1
design_analyzer>
```

ID_before

```
****************************************
Report : area
Design : pipeline
Version: v3.4a
Date   : Mon May 12 13:27:18 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              363
Number of nets:               219
Number of cells:               14
Number of references:           6

Combinational area:        333.000000
Noncombinational area:       0.000000
Net Interconnect area:     248.151855

Total cell area:           333.000000
Total area:                581.151855
1
design_analyzer>
****************************************
Report : reference
Design : pipeline
Version: v3.4a
Date   : Mon May 12 13:27:18 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO2                 lca500kv       2.000000       5     10.000000
IV                  lca500kv       1.000000       1      1.000000
IVP                 lca500kv       1.000000       1      1.000000
ND2                 lca500kv       1.000000       5      5.000000
adder                            207.000000       1    207.000000  h
```

```
decoder                        109.000000       1    109.000000  h
--------------------------------------------------------------------------------
Total 6 references                                   333.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pipeline
Version: v3.4a
Date   : Mon May 12 13:27:19 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design          Wire Loading Model      Library
------------------------------------------------
pipeline              B0.5X0.5          lca500kv
decoder               B0.5X0.5          lca500kv
adder                 B0.5X0.5          lca500kv
adder_DW01_add_30_0   B0.5X0.5          lca500kv


  Startpoint: id_pc_q[1] (input port)
  Endpoint: pc_plus_disp[29]
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                                      Incr       Path
  ------------------------------------------------------------------------------
  input external delay                                       0.00       0.00 f
  id_pc_q[1] (in)                                            0.00       0.00 f
  u3/in1[1] (adder)                                          0.00       0.00 f
  u3/add_8/A[1] (adder_DW01_add_30_0)                        0.00       0.00 f
  u3/add_8/U1_1/CO (FA1A)                                    0.63       0.63 f
  u3/add_8/U1_2/CO (FA1A)                                    0.31       0.94 f
  u3/add_8/U1_3/CO (FA1A)                                    0.31       1.25 f
  u3/add_8/U1_4/CO (FA1A)                                    0.31       1.56 f
  u3/add_8/U1_5/CO (FA1A)                                    0.31       1.87 f
  u3/add_8/U1_6/CO (FA1A)                                    0.31       2.18 f
  u3/add_8/U1_7/CO (FA1A)                                    0.31       2.49 f
  u3/add_8/U1_8/CO (FA1A)                                    0.31       2.80 f
  u3/add_8/U1_9/CO (FA1A)                                    0.31       3.11 f
  u3/add_8/U1_10/CO (FA1A)                                   0.31       3.43 f
  u3/add_8/U1_11/CO (FA1A)                                   0.31       3.74 f
  u3/add_8/U1_12/CO (FA1A)                                   0.31       4.05 f
  u3/add_8/U1_13/CO (FA1A)                                   0.31       4.36 f
  u3/add_8/U1_14/CO (FA1A)                                   0.31       4.67 f
```

```
  u3/add_8/U1_15/CO (FA1A)                                0.31       4.98 f
  u3/add_8/U1_16/CO (FA1A)                                0.31       5.29 f
  u3/add_8/U1_17/CO (FA1A)                                0.31       5.60 f
  u3/add_8/U1_18/CO (FA1A)                                0.31       5.91 f
  u3/add_8/U1_19/CO (FA1A)                                0.31       6.22 f
  u3/add_8/U1_20/CO (FA1A)                                0.31       6.53 f
  u3/add_8/U1_21/CO (FA1A)                                0.31       6.84 f
  u3/add_8/U1_22/CO (FA1A)                                0.31       7.15 f
  u3/add_8/U1_23/CO (FA1A)                                0.31       7.46 f
  u3/add_8/U1_24/CO (FA1A)                                0.31       7.77 f
  u3/add_8/U1_25/CO (FA1A)                                0.31       8.08 f
  u3/add_8/U1_26/CO (FA1A)                                0.31       8.39 f
  u3/add_8/U1_27/CO (FA1A)                                0.31       8.70 f
  u3/add_8/U1_28/CO (FA1A)                                0.33       9.03 f
  u3/add_8/U1_29/Z (EO3P)                                 0.23       9.26 r
  u3/add_8/SUM[29] (adder_DW01_add_30_0)                  0.00       9.26 r
  u3/out[29] (adder)                                      0.00       9.26 r
  pc_plus_disp[29] (out)                                  0.00       9.27 r
  data arrival time                                                  9.27
  -------------------------------------------------------------------------
-
  (Path is unconstrained)


1
design_analyzer>
```

ID_after

```
****************************************
Report : area
Design : pipeline
Version: v3.4a
Date   : Mon May 12 13:30:34 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              363
Number of nets:               219
Number of cells:               14
Number of references:           6

Combinational area:        344.000000
Noncombinational area:       0.000000
Net Interconnect area:     256.488983

Total cell area:           344.000000
Total area:                600.489014
1
design_analyzer>
****************************************
Report : reference
Design : pipeline
Version: v3.4a
Date   : Mon May 12 13:30:34 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO2                 lca500kv       2.000000       5     10.000000
IV                  lca500kv       1.000000       1      1.000000
IVP                 lca500kv       1.000000       1      1.000000
ND2                 lca500kv       1.000000       5      5.000000
adder                            207.000000       1    207.000000  h
```

```
decoder                        120.000000       1    120.000000  h
---------------------------------------------------------------------------
Total 6 references                                   344.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pipeline
Version: v3.4a
Date   : Mon May 12 13:30:35 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
pipeline              B0.5X0.5          lca500kv
decoder               B0.5X0.5          lca500kv
adder                 B0.5X0.5          lca500kv
adder_DW01_add_30_0   B0.5X0.5          lca500kv


  Startpoint: id_pc_q[1] (input port)
  Endpoint: pc_plus_disp[29]
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                                   Incr       Path
  -------------------------------------------------------------------------
  input external delay                                    0.00       0.00 f
  id_pc_q[1] (in)                                         0.00       0.00 f
  u3/in1[1] (adder)                                       0.00       0.00 f
  u3/add_8/A[1] (adder_DW01_add_30_0)                     0.00       0.00 f
  u3/add_8/U1_1/CO (FA1A)                                 0.63       0.63 f
  u3/add_8/U1_2/CO (FA1A)                                 0.31       0.94 f
  u3/add_8/U1_3/CO (FA1A)                                 0.31       1.25 f
  u3/add_8/U1_4/CO (FA1A)                                 0.31       1.56 f
  u3/add_8/U1_5/CO (FA1A)                                 0.31       1.87 f
  u3/add_8/U1_6/CO (FA1A)                                 0.31       2.18 f
  u3/add_8/U1_7/CO (FA1A)                                 0.31       2.49 f
  u3/add_8/U1_8/CO (FA1A)                                 0.31       2.80 f
  u3/add_8/U1_9/CO (FA1A)                                 0.31       3.11 f
  u3/add_8/U1_10/CO (FA1A)                                0.31       3.43 f
  u3/add_8/U1_11/CO (FA1A)                                0.31       3.74 f
  u3/add_8/U1_12/CO (FA1A)                                0.31       4.05 f
  u3/add_8/U1_13/CO (FA1A)                                0.31       4.36 f
  u3/add_8/U1_14/CO (FA1A)                                0.31       4.67 f
```

```
  u3/add_8/U1_15/CO (FA1A)                                0.31       4.98 f
  u3/add_8/U1_16/CO (FA1A)                                0.31       5.29 f
  u3/add_8/U1_17/CO (FA1A)                                0.31       5.60 f
  u3/add_8/U1_18/CO (FA1A)                                0.31       5.91 f
  u3/add_8/U1_19/CO (FA1A)                                0.31       6.22 f
  u3/add_8/U1_20/CO (FA1A)                                0.31       6.53 f
  u3/add_8/U1_21/CO (FA1A)                                0.31       6.84 f
  u3/add_8/U1_22/CO (FA1A)                                0.31       7.15 f
  u3/add_8/U1_23/CO (FA1A)                                0.31       7.46 f
  u3/add_8/U1_24/CO (FA1A)                                0.31       7.77 f
  u3/add_8/U1_25/CO (FA1A)                                0.31       8.08 f
  u3/add_8/U1_26/CO (FA1A)                                0.31       8.39 f
  u3/add_8/U1_27/CO (FA1A)                                0.31       8.70 f
  u3/add_8/U1_28/CO (FA1A)                                0.33       9.03 f
  u3/add_8/U1_29/Z (EO3P)                                 0.23       9.26 r
  u3/add_8/SUM[29] (adder_DW01_add_30_0)                  0.00       9.26 r
  u3/out[29] (adder)                                      0.00       9.26 r
  pc_plus_disp[29] (out)                                  0.00       9.27 r
  data arrival time                                                  9.27
  -----------------------------------------------------------------------
-
  (Path is unconstrained)


1
design_analyzer>
```

IS_before

```
****************************************
Report : area
Design : rob
Version: v3.4a
Date   : Tue May 13 22:54:33 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              521
Number of nets:             20947
Number of cells:            21909
Number of references:          18

Combinational area:      35045.000000
Noncombinational area:   10908.000000
Net Interconnect area:   20096.010986

Total cell area:         45953.000000
Total area:              66049.010742
1
design_analyzer>
****************************************
Report : reference
Design : rob
Version: v3.4a
Date   : Tue May 13 22:54:33 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000     333    666.000000
AN3                 lca500kv       2.000000     346    692.000000
AO2                 lca500kv       2.000000     503   1006.000000
AO3                 lca500kv       2.000000     536   1072.000000
AO222               lca500kv       4.000000     563   2252.000000
```

```
B4I                lca500kv       2.000000     434    868.000000
BUF4               lca500kv       2.000000     314    628.000000
BUF5               lca500kv       3.000000     346   1038.000000
EN                 lca500kv       3.000000     686   2058.000000
EO                 lca500kv       3.000000    1586   4758.000000
FD1Q               lca500kv       6.000000    1818  10908.000000  n
IVAP               lca500kv       2.000000     501   1002.000000
IVP                lca500kv       1.000000    3323   3323.000000
ND2                lca500kv       1.000000    4703   4703.000000
ND4                lca500kv       2.000000    2979   5958.000000
NND2               lca500kv       2.000000    1349   2698.000000
NR2                lca500kv       1.000000     855    855.000000
NR4                lca500kv       2.000000     734   1468.000000
----------------------------------------------------------------------------
Total 18 references                                 45953.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : rob
Version: v3.4a
Date   : Tue May 13 22:54:33 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
rob                   B2X2             lca500kv

  Startpoint: instr_in[0] (input port)
  Endpoint: gpr_writedata[31] (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  U222/Z (EN)                              0.35       0.35 r
  U153/Z (AN3)                             0.29       0.64 r
  U143/Z (ND4)                             0.56       1.20 f
  U30/Z (B4I)                              0.71       1.91 r
  U154/Z (NR4)                             0.36       2.27 f
  U144/Z (ND4)                             0.33       2.60 r
  U41/Z (IVAP)                             0.83       3.43 f
  U33/Z (NR2)                              0.26       3.69 r
  U36/Z (BUF4)                             0.44       4.13 r
  U147/Z (NND2)                            0.20       4.33 f
  U42/Z (BUF5)                             0.78       5.11 f
```

```
U77/Z (AO3)                                  0.31       5.42 r
U81/Z (ND2)                                  0.25       5.67 r
U23/Z (NR4)                                  0.48       6.15 f
U54/Z (IVAP)                                 0.50       6.65 r
U212/Z (AO222)                               0.58       7.23 f
U442/Z (EO)                                  0.43       7.66 r
U423/Z (EN)                                  0.38       8.04 f
U26/Z (BUF4)                                 0.44       8.48 f
U73/Z (AN3)                                  0.28       8.76 f
U163/Z (IVAP)                                0.67       9.43 r
U140/Z (ND4)                                 0.72      10.15 f
U327/Z (ND4)                                 0.68      10.83 r
U322/Z (ND2)                                 0.32      11.15 f
U362/Z (BUF4)                                0.51      11.66 r
U12/Z (IVAP)                                 0.82      12.48 f
U121/Z (NR4)                                 0.55      13.03 f
U342/Z (AN3)                                 0.37      13.40 r
U411/Z (BUF4)                                0.42      13.82 f
U35/Z (AO3)                                  0.36      14.18 r
U422/Z (BUF5)                                0.88      15.06 f
U386/Z (EO)                                  0.56      15.61 r
U264/Z (IVP)                                 0.62      16.23 f
U426/Z (NND2)                                0.32      16.55 f
U312/Z (BUF5)                                0.84      17.39 r
U88/Z (AO3)                                  0.38      17.77 f
U21/Z (ND2)                                  0.27      18.04 r
U14/Z (NR4)                                  0.49      18.53 f
U52/Z (IVAP)                                 0.61      19.14 r
gpr_writedata[31] (out)                      0.68      19.82 f
data arrival time                                      19.82
------------------------------------------------------------
(Path is unconstrained)
```

IS_after

```
****************************************
Report : area
Design : rob
Version: v3.4a
Date   : Tue May 13 22:55:43 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              526
Number of nets:             21636
Number of cells:            23226
Number of references:          28

Combinational area:      44245.000000
Noncombinational area:   11082.000000
Net Interconnect area:   21446.494012

Total cell area:         55327.000000
Total area:              76773.494012
1
design_analyzer>
****************************************
Report : reference
Design : rob
Version: v3.4a
Date   : Tue May 13 22:55:43 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000     324    648.000000
AN3                 lca500kv       2.000000     342    684.000000
AO2                 lca500kv       2.000000     451    902.000000
AO3                 lca500kv       2.000000     482    964.000000
AO222               lca500kv       4.000000     551   2204.000000
```

```
B4I                 lca500kv       2.000000     426    852.000000
BUF4                lca500kv       2.000000     307    614.000000
BUF5                lca500kv       3.000000     328    984.000000
EN                  lca500kv       3.000000     645   1935.000000
EO                  lca500kv       3.000000    1589   4767.000000
FD1Q                lca500kv       6.000000    1818  10908.000000  n
IVAP                lca500kv       2.000000     501   1002.000000
IVP                 lca500kv       1.000000    2947   2947.000000
ND2                 lca500kv       1.000000    4421   4421.000000
ND4                 lca500kv       2.000000    2895   5790.000000
NND2                lca500kv       2.000000    1349   2698.000000
NR2                 lca500kv       1.000000     844    844.000000
NR4                 lca500kv       2.000000     726   1452.000000
value_gen           lca500kv     454.000000       1    454.000000  h
guard_0             lca500kv    1182.000000       1   1182.000000  h
guard_1             lca500kv    1182.000000       1   1182.000000  h
guard_2             lca500kv    1182.000000       1   1182.000000  h
guard_3             lca500kv    1182.000000       1   1182.000000  h
guard_4             lca500kv    1182.000000       1   1182.000000  h
guard_5             lca500kv    1182.000000       1   1182.000000  h
guard_6             lca500kv    1182.000000       1   1182.000000  h
guard_7             lca500kv    1182.000000       1   1182.000000  h
shiftreg            lca500kv     801.000000       1    801.000000  h
-----------------------------------------------------------------------------
Total 28 references                                  55327.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : rob
Version: v3.4a
Date   : Tue May 13 22:55:43 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
rob                   B2X2              lca500kv
value_gen             B0.5X0.5          lca500kv
guard_0               B0.5X0.5          lca500kv
guard_1               B0.5X0.5          lca500kv
guard_2               B0.5X0.5          lca500kv
guard_3               B0.5X0.5          lca500kv
guard_4               B0.5X0.5          lca500kv
guard_5               B0.5X0.5          lca500kv
guard_6               B0.5X0.5          lca500kv
guard_7               B0.5X0.5          lca500kv
```

```
shiftreg              B0.5X0.5          lca500kv


  Startpoint: instr_in[0] (input port)
  Endpoint: gpr_writedata[31] (output port)
  Path Group: (none)
  Path Type: max

  Point                                        Incr       Path
  -----------------------------------------------------------
  U222/Z (EN)                                  0.33       0.33 r
  U153/Z (AN3)                                 0.25       0.58 r
  U143/Z (ND4)                                 0.51       1.09 f
  U30/Z (B4I)                                  0.68       1.77 r
  U154/Z (NR4)                                 0.33       2.10 f
  U144/Z (ND4)                                 0.30       2.40 r
  U41/Z (IVAP)                                 0.81       3.21 f
  U33/Z (NR2)                                  0.23       3.44 r
  U36/Z (BUF4)                                 0.54       3.98 r
  U147/Z (NND2)                                0.22       4.20 f
  U42/Z (BUF5)                                 0.56       4.76 f
  U77/Z (AO3)                                  0.36       5.12 r
  U81/Z (ND2)                                  0.23       5.35 r
  U23/Z (NR4)                                  0.43       5.78 f
  U54/Z (IVAP)                                 0.54       6.32 r
  U212/Z (AO222)                               0.52       6.84 f
  U442/Z (EO)                                  0.41       7.25 r
  U423/Z (EN)                                  0.33       7.58 f
  U26/Z (BUF4)                                 0.54       8.12 f
  U73/Z (AN3)                                  0.28       8.40 f
  U163/Z (IVAP)                                0.63       9.03 r
  U140/Z (ND4)                                 0.64       9.67 f
  U327/Z (ND4)                                 0.64      10.31 r
  U322/Z (ND2)                                 0.33      10.64 f
  U362/Z (BUF4)                                0.49      11.13 r
  U12/Z (IVAP)                                 0.82      11.95 f
  U121/Z (NR4)                                 0.55      12.50 f
  U342/Z (AN3)                                 0.37      12.87 r
  U411/Z (BUF4)                                0.46      13.33 f
  U35/Z (AO3)                                  0.46      13.79 r
  U422/Z (BUF5)                                0.76      14.55 f
  U386/Z (EO)                                  0.59      15.14 r
  U264/Z (IVP)                                 0.62      15.76 f
  U426/Z (NND2)                                0.49      16.25 f
  U312/Z (BUF5)                                0.76      17.01 r
  U88/Z (AO3)                                  0.57      17.58 f
  U21/Z (ND2)                                  0.37      17.95 r
  U14/Z (NR4)                                  0.38      18.33 f
  U52/Z (IVAP)                                 0.61      18.94 r
  u2/value_out[9] (guard_0)                    0.44      19.38 f
  gpr_writedata[31] (out)                      0.68      20.06 f
  data arrival time                                      20.06
```

```
-----------------------------------------------------------
(Path is unconstrained)
```

EX

```
****************************************
Report : area
Design : pipeline
Version: v3.4a
Date   : Mon May 12 14:13:54 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              115
Number of nets:               174
Number of cells:               71
Number of references:           2

Combinational area:       2136.000000
Noncombinational area:     420.000000
Net Interconnect area:    1925.943115

Total cell area:          2556.000000
Total area:               4481.943359
1
design_analyzer>
****************************************
Report : reference
Design : pipeline
Version: v3.4a
Date   : Mon May 12 14:13:54 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
FD1Q               lca500kv       6.000000      70    420.000000  n
alu                            2136.000000       1   2136.000000  h
-----------------------------------------------------------------------------
Total 2 references                                   2556.000000
1
```

```
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pipeline
Version: v3.4a
Date   : Mon May 12 14:13:54 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
pipeline               B2X2               lca500kv
alu                    B2X2               lca500kv
barrel_shifter         B1X1               lca500kv
adder_c                B1X1               lca500kv
add_31                 B0.5X0.5           lca500kv
add_30                 B0.5X0.5           lca500kv
add_29                 B0.5X0.5           lca500kv
add_28                 B0.5X0.5           lca500kv
add_27                 B0.5X0.5           lca500kv
cladder_9              B0.5X0.5           lca500kv
cladder_8              B0.5X0.5           lca500kv
add_26                 B0.5X0.5           lca500kv
add_25                 B0.5X0.5           lca500kv
add_24                 B0.5X0.5           lca500kv
cladder_7              B0.5X0.5           lca500kv
add_23                 B0.5X0.5           lca500kv
add_22                 B0.5X0.5           lca500kv
cladder_6              B0.5X0.5           lca500kv
add_21                 B0.5X0.5           lca500kv
add_20                 B0.5X0.5           lca500kv
add_19                 B0.5X0.5           lca500kv
add_18                 B0.5X0.5           lca500kv
add_17                 B0.5X0.5           lca500kv
cladder_5              B0.5X0.5           lca500kv
add_16                 B0.5X0.5           lca500kv
add_15                 B0.5X0.5           lca500kv
add_14                 B0.5X0.5           lca500kv
cladder_4              B0.5X0.5           lca500kv
add_13                 B0.5X0.5           lca500kv
add_12                 B0.5X0.5           lca500kv
add_11                 B0.5X0.5           lca500kv
add_10                 B0.5X0.5           lca500kv
cladder_3              B0.5X0.5           lca500kv
add_9                  B0.5X0.5           lca500kv
add_8                  B0.5X0.5           lca500kv
add_7                  B0.5X0.5           lca500kv
```

```
add_6                B0.5X0.5        lca500kv
cladder_2            B0.5X0.5        lca500kv
add_5                B0.5X0.5        lca500kv
add_4                B0.5X0.5        lca500kv
add_3                B0.5X0.5        lca500kv
add_2                B0.5X0.5        lca500kv
cladder_1            B0.5X0.5        lca500kv
cladder1             B0.5X0.5        lca500kv
add_1                B0.5X0.5        lca500kv
cladder_0            B0.5X0.5        lca500kv
add_0                B0.5X0.5        lca500kv


  Startpoint: ex_decode_q_reg[8]
              (rising edge-triggered flip-flop)
  Endpoint: ex_result_d[0]
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  ex_decode_q_reg[8]/CP (FD1Q)             0.00       0.00 r
  ex_decode_q_reg[8]/Q (FD1Q)              0.53       0.53 f
  r4/control[8] (alu)                      0.00       0.53 f
  r4/U37/Z (NR2)                           0.20       0.73 r
  r4/U38/Z (BUF4)                          0.55       1.28 r
  r4/U48/Z (B1A)                           0.84       2.12 f
  r4/U117/Z (AO2)                          0.39       2.51 r
  r4/u2/operand2[1] (adder_c)              0.00       2.51 r
  r4/u2/U112/Z (BUF1)                      0.41       2.92 r
  r4/u2/U73/Z (EO1)                        0.80       3.72 r
  r4/u2/U39/Z (NNR2)                       0.36       4.09 f
  r4/u2/r11/g[1] (cladder_7)               0.00       4.09 f
  r4/u2/r11/U11/Z (AO21)                   0.28       4.37 f
  r4/u2/r11/U12/Z (AO21)                   0.30       4.67 f
  r4/u2/r11/U8/Z (AO21)                    0.43       5.10 f
  r4/u2/r11/go (cladder_7)                 0.00       5.10 f
  r4/u2/r3/g[0] (cladder_8)                0.00       5.10 f
  r4/u2/r3/U11/Z (AO21)                    0.30       5.40 f
  r4/u2/r3/U12/Z (AO21)                    0.30       5.70 f
  r4/u2/r3/U8/Z (AO21)                     0.38       6.08 f
  r4/u2/r3/go (cladder_8)                  0.00       6.08 f
  r4/u2/r1/g[0] (cladder1)                 0.00       6.08 f
  r4/u2/r1/U3/Z (AO21)                     0.38       6.46 f
  r4/u2/r1/cout[0] (cladder1)              0.00       6.46 f
  r4/u2/r2/cin (cladder_5)                 0.00       6.46 f
  r4/u2/r2/U6/Z (AO21)                     0.40       6.86 f
  r4/u2/r2/U7/Z (AO21)                     0.48       7.34 f
  r4/u2/r2/U9/Z (AO21)                     0.36       7.70 f
  r4/u2/r2/cout[2] (cladder_5)             0.00       7.70 f
  r4/u2/r4/cin (cladder_9)                 0.00       7.70 f
```

```
  r4/u2/r4/U6/Z (AO21)                        0.36       8.06 f
  r4/u2/r4/U7/Z (AO21)                        0.43       8.49 f
  r4/u2/r4/U9/Z (AO21)                        0.36       8.85 f
  r4/u2/r4/cout[2] (cladder_9)                0.00       8.85 f
  r4/u2/r12/cin (add_6)                       0.00       8.85 f
  r4/u2/r12/U3/Z (EO3)                        0.63       9.48 r
  r4/u2/r12/out (add_6)                       0.00       9.48 r
  r4/u2/neg (adder_c)                         0.00       9.48 r
  r4/U198/Z (NNR2)                            0.29       9.77 r
  r4/U199/Z (AO2)                             0.22       9.99 f
  r4/U80/Z (AO3)                              0.16      10.15 r
  r4/result[0] (alu)                          0.00      10.15 r
  ex_result_d[0] (out)                        0.00      10.15 r
  data arrival time                                     10.15
  -----------------------------------------------------------
  (Path is unconstrained)


1
design_analyzer>
```

MEM

```
****************************************
Report : area
Design : pipeline
Version: v3.4a
Date   : Mon May 12 14:29:11 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              197
Number of nets:               164
Number of cells:               65
Number of references:           1

Combinational area:          0.000000
Noncombinational area:     390.000000
Net Interconnect area:     111.815491

Total cell area:           390.000000
Total area:                501.815491
1
design_analyzer>
****************************************
Report : reference
Design : pipeline
Version: v3.4a
Date   : Mon May 12 14:29:11 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
FD1Q                lca500kv       6.000000      65    390.000000  n
-----------------------------------------------------------------------------
Total 1 references                                     390.000000
1
design_analyzer>
```

```
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : pipeline
Version: v3.4a
Date   : Mon May 12 14:29:12 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
pipeline               B0.5X0.5          lca500kv


  Startpoint: dcache_hit (input port)
  Endpoint: cache_hit_out
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                    13.80      13.80 r
  dcache_hit (in)                          0.00      13.80 r
  cache_hit_out (out)                      0.00      13.80 r
  data arrival time                                  13.80
  -----------------------------------------------------------
  (Path is unconstrained)


1
design_analyzer>
```

The Value Field Generator

```
****************************************
Report : area
Design : value_gen
Version: v3.4a
Date   : Wed Apr 30 02:43:24 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              172
Number of nets:               199
Number of cells:               46
Number of references:           9

Combinational area:        454.000000
Noncombinational area:       0.000000
Net Interconnect area:     360.835541

Total cell area:           454.000000
Total area:                814.835571
1
design_analyzer>
****************************************
Report : reference
Design : value_gen
Version: v3.4a
Date   : Wed Apr 30 02:43:25 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AO6                 lca500kv       2.000000       9     18.000000
AO21N               lca500kv       2.000000       9     18.000000
AO222               lca500kv       4.000000      21     84.000000
BUF4                lca500kv       2.000000       2      4.000000
IVP                 lca500kv       1.000000       1      1.000000
```

```
NNR2               lca500kv       2.000000       1      2.000000
NR2P               lca500kv       2.000000       1      2.000000
OA21               lca500kv       2.000000       1      2.000000
mask_gen                        323.000000       1    323.000000  h
----------------------------------------------------------------------------
Total 9 references                                    454.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : value_gen
Version: v3.4a
Date   : Wed Apr 30 02:43:25 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
value_gen              B1X1              lca500kv
mask_gen               B0.5X0.5          lca500kv
guarddec_3             B0.5X0.5          lca500kv
guarddec_2             B0.5X0.5          lca500kv
guarddec_1             B0.5X0.5          lca500kv
guarddec_0             B0.5X0.5          lca500kv


  Startpoint: reg_data[28]
              (input port)
  Endpoint: value_out[28]
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                     0.00       0.00 f
  reg_data[28] (in)                        0.00       0.00 f
  r1/guard_reg[28] (mask_gen)              0.00       0.00 f
  r1/U76/Z (NR4)                           0.54       0.54 r
  r1/U29/Z (ND8P)                          0.98       1.52 f
  r1/U97/Z (EN)                            0.33       1.85 r
  r1/U63/Z (OA21)                          0.71       2.56 r
  r1/U30/Z (ND2)                           0.28       2.84 f
  r1/bit_mask[20] (mask_gen)               0.00       2.84 f
  U23/Z (AO222)                            0.52       3.36 f
  value_out[28] (out)                      0.00       3.36 f
  data arrival time                                   3.36
```

```
  ----------------------------------------------------------
  (Path is unconstrained)


1
design_analyzer>
```

The Control Logic

```
****************************************
Report : area
Design : guard
Version: v3.4a
Date   : Wed Apr 30 03:21:17 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:              152
Number of nets:               316
Number of cells:              147
Number of references:          22

Combinational area:       1062.000000
Noncombinational area:     120.000000
Net Interconnect area:     948.418884

Total cell area:          1182.000000
Total area:               2130.418945
1
design_analyzer>
****************************************
Report : reference
Design : guard
Version: v3.4a
Date   : Wed Apr 30 03:21:17 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000       1      2.000000
AN2P                lca500kv       2.000000       1      2.000000
AO2                 lca500kv       2.000000      20     40.000000
BUF4                lca500kv       2.000000       1      2.000000
BUF5                lca500kv       3.000000       1      3.000000
```

```
EN                   lca500kv       3.000000       9     27.000000
EO1                  lca500kv       3.000000      32     96.000000
IV                   lca500kv       1.000000      20     20.000000
IVA                  lca500kv       1.000000       1      1.000000
IVAP                 lca500kv       2.000000       1      2.000000
ND2                  lca500kv       1.000000      20     20.000000
ND2P                 lca500kv       2.000000       1      2.000000
ND8                  lca500kv       6.000000       1      6.000000
NND2P                lca500kv       3.000000       1      3.000000
NNR2                 lca500kv       2.000000       1      2.000000
NR2                  lca500kv       1.000000       1      1.000000
NR3P                 lca500kv       3.000000       4     12.000000
NR4                  lca500kv       2.000000       8     16.000000
OR3                  lca500kv       2.000000      20     40.000000
shifter                           413.000000       1    413.000000  h
sorter                            102.000000       1    102.000000  h
tfshifter                         370.000000       1    370.000000  h, n
----------------------------------------------------------------------------
Total 22 references                                    1182.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : guard
Version: v3.4a
Date   : Wed Apr 30 03:21:17 1997
****************************************

Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design           Wire Loading Model      Library
------------------------------------------------
guard                 B1X1               lca500kv
shifter               B1X1               lca500kv
tfshifter             B0.5X0.5           lca500kv
sorter                B0.5X0.5           lca500kv


  Startpoint: op_data_in[28]
              (input port)
  Endpoint: value_out[9]
            (output port)
  Path Group: (none)
  Path Type: max

  Point                                      Incr       Path
  -----------------------------------------------------------
  input external delay                       0.00       0.00 f
```

```
  op_data_in[28] (in)                         0.00       0.00 f
  U137/Z (NR4)                                0.56       0.56 r
  U84/Z (ND8)                                 0.66       1.22 f
  U88/Z (EN)                                  0.64       1.86 r
  U40/Z (NR3P)                                0.38       2.24 f
  U34/Z (IVA)                                 0.42       2.66 r
  U60/Z (ND2)                                 0.49       3.15 f
  r1/value_in[23] (shifter)                   0.00       3.15 f
  r1/U148/Z (AO2)                             0.35       3.50 r
  r1/U226/Z (AO3)                             0.47       3.97 f
  r1/U127/Z (NNR2)                            0.24       4.21 f
  r1/U96/Z (AO2)                              0.44       4.66 r
  r1/U86/Z (NR2)                              0.22       4.88 f
  r1/U87/Z (OA21)                             0.32       5.20 f
  r1/U152/Z (AO1)                             0.47       5.67 r
  r1/U84/Z (AO3)                              0.27       5.94 f
  r1/value_out[9] (shifter)                   0.00       5.94 f
  value_out[9] (out)                          0.00       5.94 f
  data arrival time                                      5.94
  -----------------------------------------------------------
  (Path is unconstrained)


1
design_analyzer>
```

The Shift Register

```
****************************************
Report : area
Design : shiftreg
Version: v3.4a
Date   : Wed Apr 30 03:53:09 1997
****************************************

Library(s) Used:

    lca500kv (File: /p/cad/lsitk-3.0/lib/synopsys/lca500k/lca500kv.db)

Number of ports:               63
Number of nets:               388
Number of cells:              321
Number of references:          43

Combinational area:        627.000000
Noncombinational area:     174.000000
Net Interconnect area:     592.334961

Total cell area:           801.000000
Total area:               1393.334961
1
design_analyzer>
****************************************
Report : reference
Design : shiftreg
Version: v3.4a
Date   : Wed Apr 30 03:53:10 1997
****************************************

Attributes:
    b - black box (unknown)
   bo - allows boundary optimization
    d - dont_touch
   mo - map_only
    h - hierarchical
    n - noncombinational
    r - removable
    s - synthetic operator
    u - contains unmapped logic

Reference          Library       Unit Area   Count    Total Area   Attributes
-----------------------------------------------------------------------------
AN2                 lca500kv       2.000000       1      2.000000
AN2P                lca500kv       2.000000       1      2.000000
AO1                 lca500kv       2.000000       5     10.000000
AO2                 lca500kv       2.000000      46     92.000000
AO3                 lca500kv       2.000000      38     76.000000
```

```
AO4               lca500kv      2.000000      4      8.000000
AO4P              lca500kv      4.000000      1      4.000000
AO6               lca500kv      2.000000      2      4.000000
AO7               lca500kv      2.000000     11     22.000000
AO7P              lca500kv      3.000000      1      3.000000
AO21              lca500kv      2.000000      2      4.000000
AO22P             lca500kv      3.000000      2      6.000000
AO211             lca500kv      3.000000      1      3.000000
AO222             lca500kv      4.000000      4     16.000000
BUF1              lca500kv      1.000000      5      5.000000
BUF4              lca500kv      2.000000      1      2.000000
EO1               lca500kv      3.000000      5     15.000000
EON1              lca500kv      3.000000      1      3.000000
FD1Q              lca500kv      6.000000     29    174.000000  n
IV                lca500kv      1.000000     23     23.000000
IVA               lca500kv      1.000000     15     15.000000
IVAP              lca500kv      2.000000      2      4.000000
IVP               lca500kv      1.000000     16     16.000000
ND2               lca500kv      1.000000     30     30.000000
ND2P              lca500kv      2.000000      2      4.000000
ND3               lca500kv      2.000000      1      2.000000
ND4               lca500kv      2.000000      4      8.000000
ND4P              lca500kv      4.000000      1      4.000000
NND2              lca500kv      2.000000      2      4.000000
NNR2              lca500kv      2.000000      2      4.000000
NR2               lca500kv      1.000000     19     19.000000
NR2P              lca500kv      2.000000      3      6.000000
NR3               lca500kv      2.000000      2      4.000000
NR4               lca500kv      2.000000      4      8.000000
OA21              lca500kv      2.000000     25     50.000000
OR2P              lca500kv      2.000000      1      2.000000
OR3               lca500kv      2.000000      3      6.000000
onepos_0                        7.000000      1      7.000000  h
onepos_1                        8.000000      1      8.000000  h
onepos_2                        8.000000      1      8.000000  h
onepos_3                        8.000000      1      8.000000  h
onepos_4                        8.000000      1      8.000000  h
sorter                        102.000000      1    102.000000  h
-----------------------------------------------------------------------------
Total 43 references                                801.000000
1
design_analyzer>
****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : shiftreg
Version: v3.4a
Date   : Wed Apr 30 03:53:10 1997
****************************************
```

```
Operating Conditions: NOM   Library: lca500kv
Wire Loading Model Mode: enclosed

Design          Wire Loading Model      Library
------------------------------------------------
shiftreg              B1X1              lca500kv
onepos_4              B0.5X0.5          lca500kv
onepos_3              B0.5X0.5          lca500kv
onepos_2              B0.5X0.5          lca500kv
onepos_1              B0.5X0.5          lca500kv
sorter                B0.5X0.5          lca500kv
onepos_0              B0.5X0.5          lca500kv


  Startpoint: modify_in (input port)
  Endpoint: count[1] (output port)
  Path Group: (none)
  Path Type: max

  Point                                    Incr       Path
  -----------------------------------------------------------
  input external delay                     0.00       0.00 r
  modify_in (in)                           0.00       0.00 r
  U106/Z (IVAP)                            0.50       0.50 f
  U155/Z (OA21)                            0.60       1.10 f
  U145/Z (NR4)                             0.49       1.59 r
  U91/Z (BUF1)                             0.38       1.97 r
  U337/Z (IV)                              0.23       2.20 f
  U203/Z (NR2)                             0.19       2.39 r
  U146/Z (ND4)                             0.40       2.79 f
  U338/Z (IVA)                             0.22       3.01 r
  U147/Z (ND4)                             0.60       3.62 f
  U205/Z (NR2)                             0.23       3.85 r
  U148/Z (ND4)                             0.30       4.15 f
  U95/Z (BUF1)                             0.49       4.64 f
  U269/Z (IVA)                             0.32       4.96 r
  U206/Z (AO1)                             0.25       5.21 f
  U149/Z (AN2)                             0.28       5.49 f
  U100/Z (ND4P)                            0.33       5.82 r
  count[1] (out)                           0.00       5.82 r
  data arrival time                                   5.82
  -----------------------------------------------------------
  (Path is unconstrained)


1
design_analyzer>
```