SAN JOSESTATEUNIVERSITY

CharlesW.DavidsonCollege of Engineering

DEPARTMENT OF ELECTRICAL ENGINEERING

**EE271 – Advanced Digital System Design and Synthesis**

Fall 2014 Final Project Report

A Simple 8-bit Scalar Processor

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Praveen Waliitagi | **Email** | [Praveen.waliitagi@sjsu.edu](mailto:Praveen.waliitagi@sjsu.edu) |
| **SJSUID** | 010059257 | **Phone** | 6694009013 |

Table of Contents

[Executive Summary: 3](#_Toc405926429)

[I. GeneralProject Information 3](#_Toc405926430)

[Table I.1: List of EDA Tools Used 3](#_Toc405926431)

[Table I.2: List of Libraries Used 3](#_Toc405926432)

[Table I.3: List of Verilog Modules (both design and test modules) 3](#_Toc405926433)

[II. The Design Overview 6](#_Toc405926434)

[Table II.1: Number of Clock Cycles Required for Each Instruction 8](#_Toc405926435)

[II.2: Experiments to obtain optimum number of clock cycles 8](#_Toc405926436)

[III. RTL-LevelSimulations/Tests for Each Instruction 9](#_Toc405926437)

[IV. RTL-Level Simulations/Tests of Whole Design (processor) 13](#_Toc405926438)

[IV.1 Test Program, Machine Codes, and Test Data 13](#_Toc405926439)

[Table IV.1: Memory Code Address (in Hex), Test Instruction, and Machine Code (in Hex) 13](#_Toc405926440)

[Table IV.2: Initial Test Data Stored atMemory Addresses (in Hex) 14](#_Toc405926441)

[IV.2 Test Results 14](#_Toc405926442)

[V. Synthesis and Optimizations of Whole Design (processor) 15](#_Toc405926443)

[Table V.1: Synthesis Constraints and Results 15](#_Toc405926444)

[V.III Relationship between Area Power and operating clock frequency 16](#_Toc405926445)

[VI. Gate-Level Simulations/Tests of Whole Design (processor) 17](#_Toc405926446)

[VII. Conclusion 18](#_Toc405926447)

[VIII. References 18](#_Toc405926448)

[Appendix A 19](#_Toc405926449)

[A.1.1 RTL simulation command 19](#_Toc405926450)

[A.2.1 Synthesis\_script 19](#_Toc405926451)

[Completed Verilog Source Codes and Test benches 22](#_Toc405926452)

[Reports and Circuits from EDA Tools 46](#_Toc405926453)

[C.1 Reports (contents) from RTL (Pre-synthesis) Simulations (VCS or NCVERILOG) 46](#_Toc405926454)

[C.2 Reports (contents) from Netlist (Post-synthesis) Simulations (VCS or NCVERILOG) 48](#_Toc405926455)

[C.3 Reports (contents) from Synthesis (Design Compiler) 52](#_Toc405926456)

[C.3.1 Area Report 52](#_Toc405926457)

[C.3.2 Power Report 53](#_Toc405926458)

[C.3.3 Timing Report 54](#_Toc405926459)

[C.4 Screenshot Circuits from Synthesis (Design Compiler) 61](#_Toc405926460)

[C.4.1 TopLevel Schematic of the design 61](#_Toc405926461)

[C.4.2 Controller Schematic 61](#_Toc405926462)

[C.4.3 Data\_Block Schematic 62](#_Toc405926463)

# Executive Summary:

In this Project an 8 bit Scalar processor is designed and implemented in verilog HDL. RTL level simulations and Gate level simulations were carried out to test the functionality of the processor. After successful completion of RTL level simulations, synthesis was performed using design compiler to obtain gate level netlist. Timing analysis was done using “update\_timing” and “report\_timing” commands in design\_compiler. Synthesis was performed by setting various values for design constraints to obtain a better tradeoff between area, power and speed. Simulation and synthesis test results are described in detail in this project report.

# I. GeneralProject Information

## Table I.1: List of EDA Tools Used

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Company** | **Used for** | **Free? (Y/N)** | **Software Documents** |
| VCS | Synopsys | Simulation & test | No |  |
| NCSIM | Cadence | Simulation & test | No |  |
| Design Vision | Synopsys | Synthesis | No |  |

## Table I.2: List of Libraries Used

|  |  |  |  |
| --- | --- | --- | --- |
| **Library file name** | **Company** | **Used with**  (EDA tool) | **The libraries are at**  (directories on eecad systems) |
| tc240c.db\_BCCOM25 | Toshiba | Synopsys | /apps/toshiba/sjsu/synopsys/tc240c/ |
| tc240c.db\_WCCOM25 | Toshiba | Synopsys | /apps/toshiba/sjsu/synopsys/tc240c/ |
|  |  |  |  |

## Table I.3: List of Verilog Modules (both design and test modules)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module Name** | **Input Ports** | **Output Ports** | **BiDirectional** | **Short Description** |
| s8sp | clk;  reset; | add;  wrt;  rd; | dat; | Top level design of the processor |
| Data\_block | clk;  reset;  load\_ar;  ar\_on\_addr;  ar\_2\_data;  load\_dr;  dr\_2\_data;  load\_lsb\_gr;  load\_msb\_gr;  gr\_2\_data;  load\_ar\_2\_pr;  inc\_pr;  pr\_2\_data;  pr\_on\_addr;  load\_ir;  ir\_2\_data;  sub\_nadd;  add\_oprnd1\_sel;  add\_oprnd2\_sel;  alu\_2\_data;  flag\_2\_data; | addr\_bus;  ir\_code; | data\_bus; | Data Handling Block where all the data manipulations occur. This block contains all the registers and ALU. |
| sys\_controller | clk;  reset;  ctrl\_ir\_code; | rd\_mem;  wr\_mem;  ctrl\_load\_ar;  ctrl\_ar\_on\_addr;  ctrl\_ar\_2\_data;  ctrl\_load\_dr;  ctrl\_dr\_2\_data;  ctrl\_load\_lsb\_gr;  ctrl\_load\_msb\_gr;  ctrl\_gr\_2\_data;  ctrl\_load\_ar\_2\_pr;  ctrl\_inc\_pr;  ctrl\_pr\_2\_data;  ctrl\_pr\_on\_addr;  ctrl\_load\_ir;  ctrl\_ir\_2\_data;  ctrl\_alu\_2\_data;  ctrl\_sub\_nadd;  ctrl\_add\_oprnd1\_sel;  ctrl\_add\_oprnd2\_sel;  ctrl\_flag\_2\_data; |  | Controller block sends control signals to data block for data manipulations. This Block reads the instruction from the instruction code register and then sends the control signals accordingly. |
| program\_cnt\_reg | clk;  reset;  inc\_pr;  load\_ar\_2\_pr;  data\_on\_pr; | pr\_on\_bus; |  | Program counter register which Points to the location in the memory from where next instruction is to be fetched |
| data\_reg | clk;  reset;  data\_on\_dr;  load\_dr | dr\_on\_data; |  | Data Register Block used data manipulation. |
| gen\_reg | clk;  reset;  load\_lsb\_gr;  load\_msb\_gr;  data\_on\_gr; | gr\_on\_data; |  | General Register Block used data manipulation. |
| addr\_reg | clk;  reset;  data\_on\_ar;  load\_ar | ar\_on\_bus; |  | Address Register used to hold the memory address and it can also be used for data manupulations |
| local\_ir\_reg | clk;  reset;  load\_ir;  data\_on\_ir; | ir\_out; |  | Instruction register used to hold the instruction fetched from memory. |
| flag\_reg | clk;  reset;  alu\_2\_data;  alu\_data; | flag\_reg; |  | Flag Register which holds the status of various arithmetic operations. |
| **Module Name** | **Input Ports** | **Output Ports** | **BiDirectional** | **Short Description** |
| alu\_block | reg\_in1;  reg\_in2;  sub\_nadd; | alu\_out; |  | Top level block of Arithmetic Logic Unit used to per Addition and subtraction |
| CLA8bit | in1;  in2;  cin; | sum;  cout; |  | 8 Bit carry Look ahead adder. Formed by instantiating two 4 bit Carry lookahead circuit. |
| CLA\_N\_bit | in1;  in2;  cin; | sum;  cout; |  | Generic N bit Carry Lookahead adder module. |
| **Testbench Module** | | | | |
| **TB\_name** | **Wire** | **Reg** |  | **Description** |
| tb\_s8sp | add;  wrt;  rd;  dat; | Clk;  Reset;  Out\_data;  mem |  | Generic Testbench which can be used to test both RTL and Netlist. |

# 

# II. The Design Overview

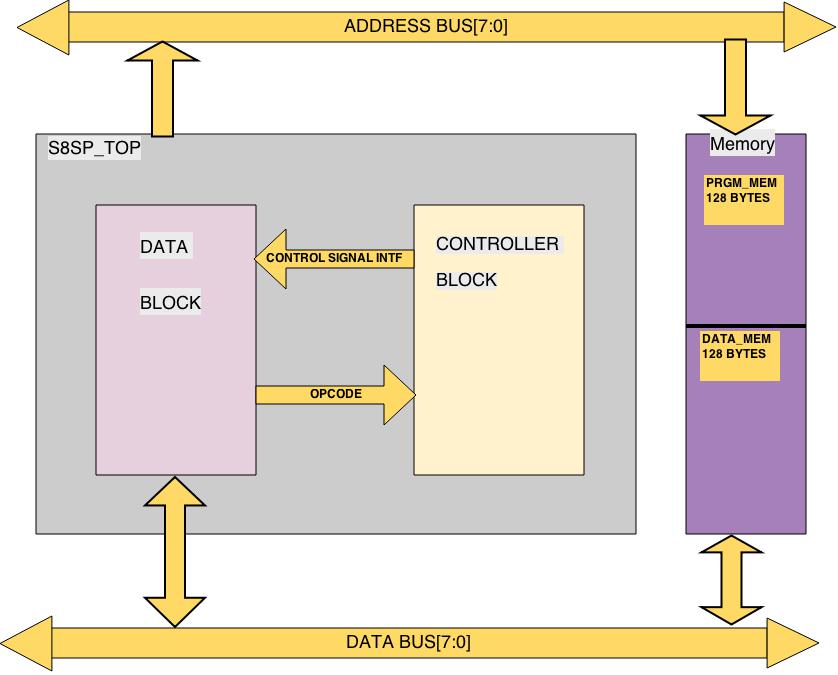
The RTL level design of small 8 bit scalar processor has data processing unit and control unit.

**DATA PROCESSING UNIT:** consists of all the registers and ALU block. As per the design specification we have the following set of registers 1. Address register (AR) which holds the memory address from where the data has to be read/write. 2. Data register (DR) which is used for data manipulations. 3. Genral register (GR) which is used to load the data read from the memory. 4. Program counter (PC) register which holds the memory address from where the next instruction has to be fetched. 5. Flag register (FR) which holds the conditions of arithmetic operations. All these registers except the flag registers are multiplexed at the input of ALU so that they can be used as source or destination in the instruction code. The ALU unit consists of a generic N bit carry look-ahead adder circuit. This generic adder is parameterized to two 4 bit carry look-ahead adder circuit to form 8 bit CLA circuit. This adder circuit is intern used for subtraction by using the necessary EXOR logic. There is also an internal instruction code register (IR) which holds the instruction fetched from the memory.

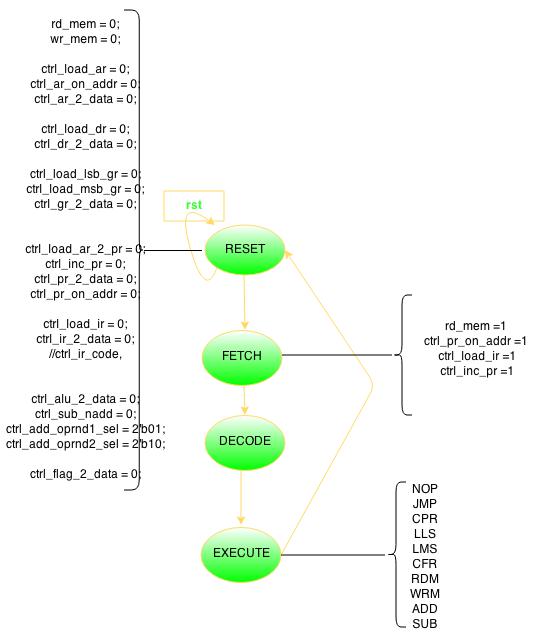
**CONTROL UNIT:**  Controller unit of the 8 bit scalar processor has 4 states namely reset, fetch, decode, and execute. In reset state all the registers and control signals values are set to zero. In fetch state all the signals necessary to fetch the instruction from the memory are asserted and PC value is incremented by one so as to be ready to fetch the next instruction from memory. In decode state IR value is read and then the state moves to execute. In execution phase based on the kind of instruction necessary control signals are asserted in order to perform the necessary operation. Memory is implemented in testbench.

**Figure II.1**: Overall Processor Block Diagram

Simplified processor block diagram is as shown below. Data block contains all the register modules and the ALU. Controller block contains a state machine which sequences the operation based on the fetched instruction. Memory is implemented in the testbench and is not part of the design



**Figure II.1**: Overall Processor Block Diagram



**Figure II.2**: Overall State Transition Diagram

## Table II.1: Number of Clock Cycles Required for Each Instruction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instruction** | **# of Clock Cycles** | **Instruction** | **# of Clock Cycles** | **Instruction** | **# of Clock Cycles** |
| NOP | 1 | LMS | 1 | ADD | 2 |
| JMP | 1 | CFR | 1 | SUB | 2 |
| CPR | 1 | RDM | 2 |  |  |
| LLS | 1 | WRM | 2 |  |  |

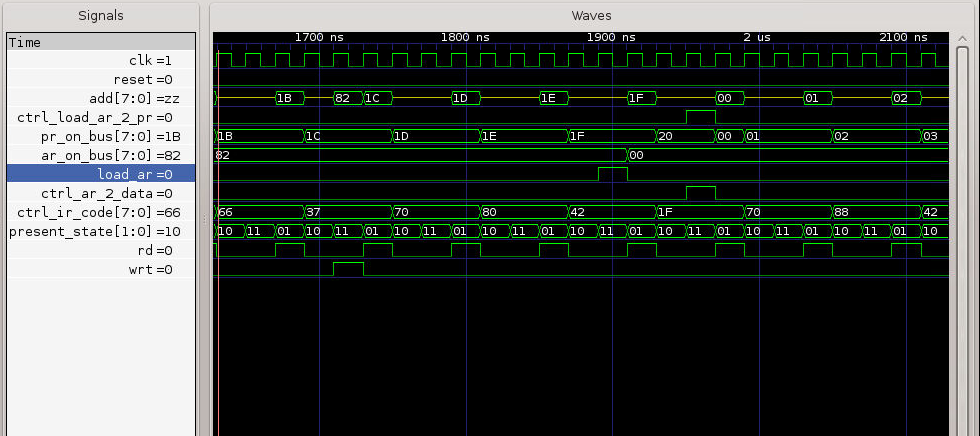
## II.2: Experiments to obtain optimum number of clock cycles

To get the best results for Arithmetic instructions ALU block was designed and tested independently. Single 4 bit carry-look-ahead (CLA) adder circuit was designed and cascaded to obtain an 8 bit CLA circuit. Synthesis of 8 bit CLA circuit was carried out and timing analysis was done to obtain the maximum delay in the circuit. This 8 bit CLA adder circuit was used for SUB instruction by using an additional EXOR logic for the 1st carry-in and to one of the inputs.

To get the best results for RDM and WRM instructions synchronous and asynchronous memory was implemented in test bench. Since memory was part of the test bench and is not synthesized asynchronous memory was used to get the faster response.

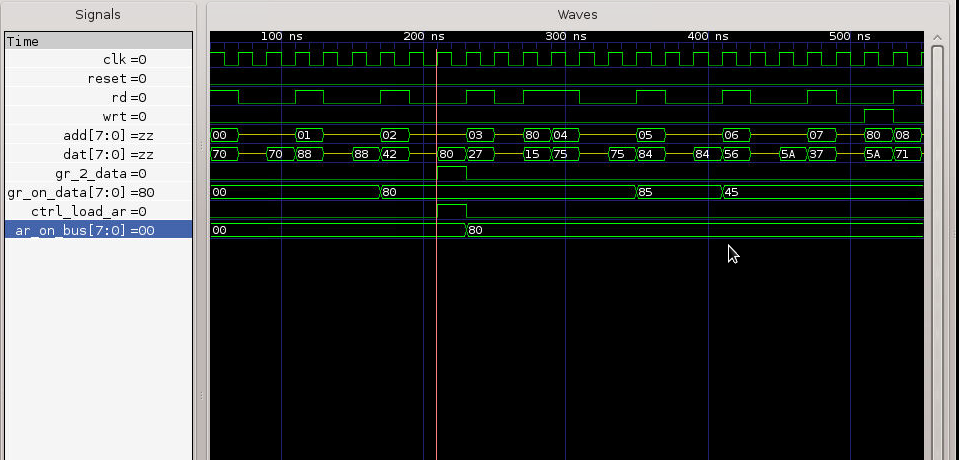
# III. RTL-LevelSimulations/Tests for Each Instruction

In the below waveform JMP instruction is executed in which AR register value is loaded into PR so as to fetch the next instruction from the JMP address. Corresponding control signals are shown in the waveform.



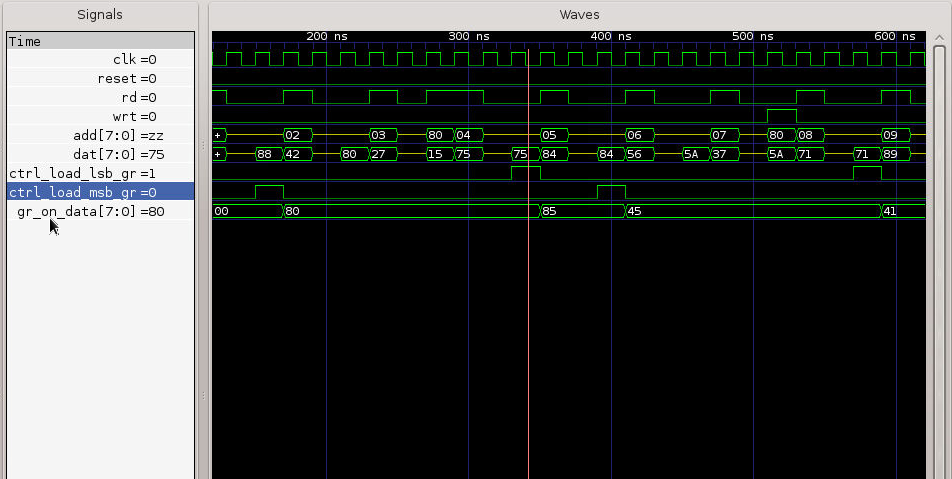
**Figure III.1**: Test Result for **JMP** Instruction

Waveform below shows copy of GR register to AR register. gr\_2\_data signal is asserted and load\_ar signal is asserted.



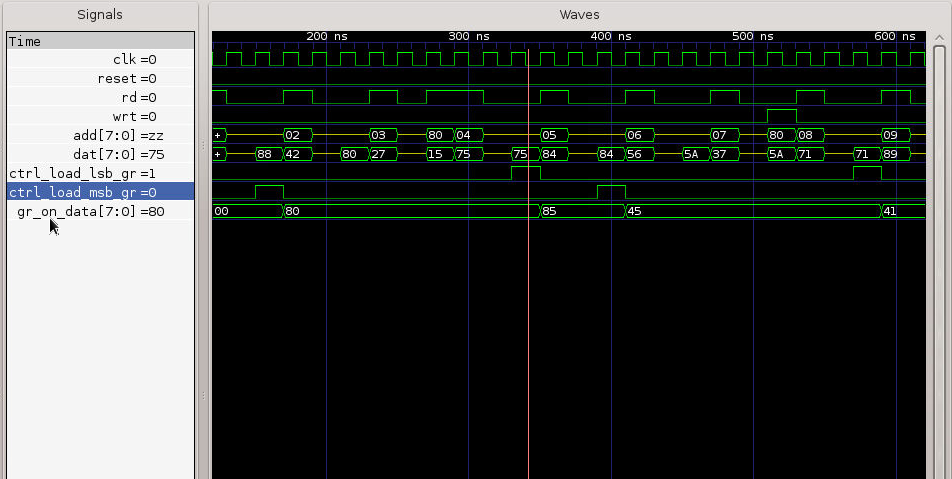
**Figure III.2**: Test Result for **CPR** Instruction

In the below waveform Last four significant bits of GR register is loaded with a value on the data bus. Ctrl\_load\_lsb\_gr is asserted to 1.



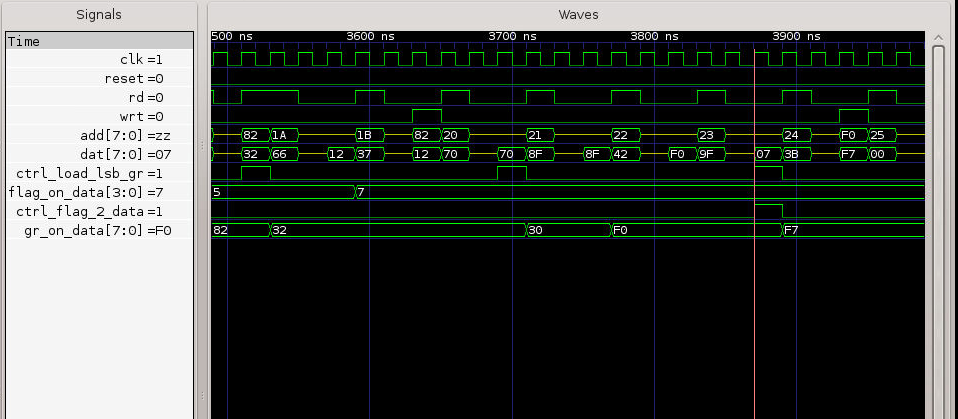
**Figure III.3**: Test Result for **LLS** Instruction

In the below waveform four most significant bits of GR register is loaded with a value on the data bus. Ctrl\_load\_msb\_gr is asserted to 1.



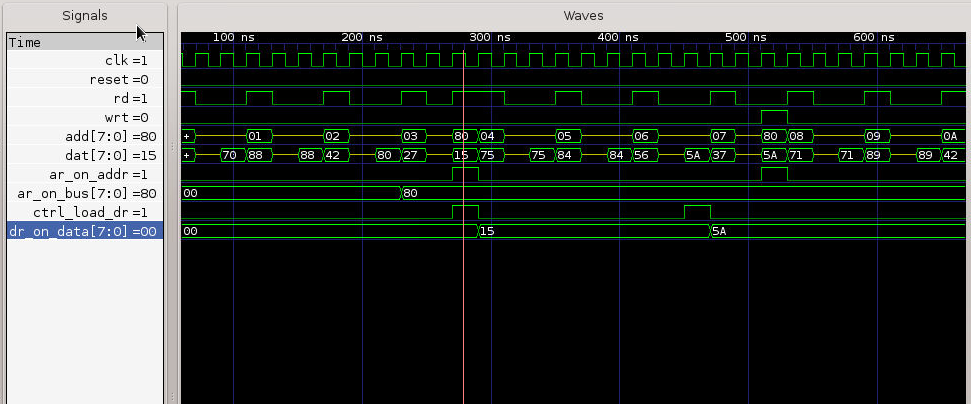
**Figure III.4**: Test Result for **LMS** Instruction

In the below waveform, contents of the flag register are copied to least significant bits of general register. control signals ctrl\_flag\_2\_data and ctrl\_load\_lsb\_gr are set to 1.



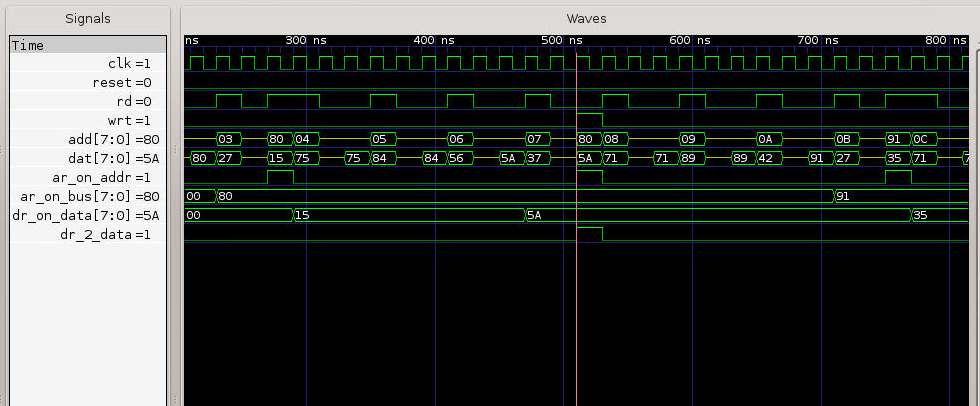
**Figure III.5**: Test Result for **CFR** Instruction

Below waveform shows AR register value loaded onto address bus to fetch the data from that memory address and also ctrl\_load\_dr set to one to load the fetched value onto DR register.



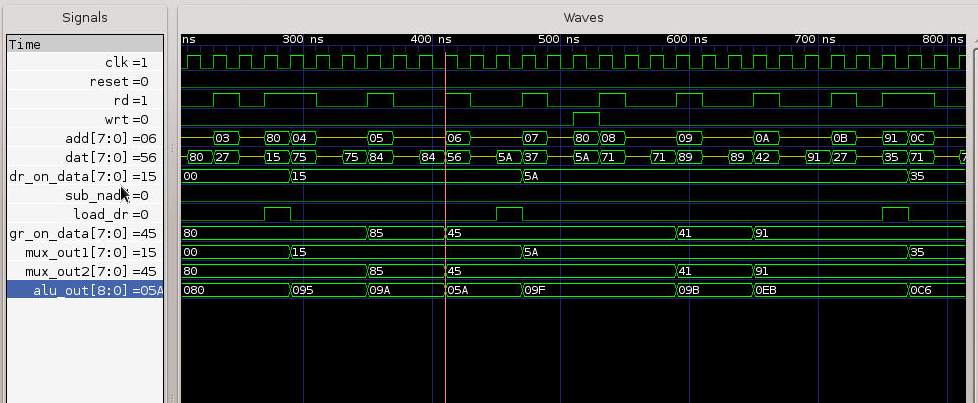
**Figure III.6**: Test Result for **RDM** Instruction

Below waveform shows AR register value loaded onto address bus to write the data from DR register onto the memory address of AR.

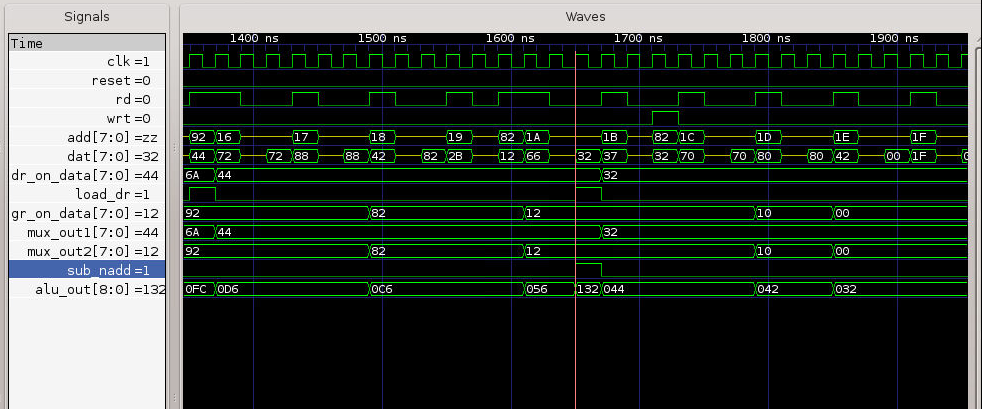


**Figure III.7**: Test Result for **WRM** Instruction

Below waveform shows Addition of contents of DR and GR register and the result getting stored in DR register.



**Figure III.8**: Test Result for **ADD** Instruction

Below waveform shows subtraction of contents of DR and GR register and the result getting stored in DR register. 

**Figure III.9**: Test Result for **SUB** Instruction

# IV. RTL-Level Simulations/Tests of Whole Design (processor)

## IV.1 Test Program, Machine Codes, and Test Data

## Table IV.1: Memory Code Address (in Hex), Test Instruction, and Machine Code (in Hex)

|  |  |  |
| --- | --- | --- |
| **Memory Address** | **Instruction** | **Machine Code** |
| 00 | LLS 0 | 70 |
| 01 | LMS 8 | 88 |
| 02 | CPR AR GR | 42 |
| 03 | RDM DR | 27 |
| 04 | LLS 5 | 75 |
| 05 | LMS 4 | 84 |
| 06 | ADD DR GR | 56 |
| 07 | WRM DR | 37 |
| 08 | LLS 1 | 71 |
| 09 | LMS 9 | 89 |
| 0A | CPR AR GR | 42 |
| 0B | RDM DR | 27 |
| 0C | LLS 1 | 71 |
| 0D | LMS 8 | 88 |
| 0E | CPR AR GR | 42 |
| 0F | RDM GR | 2B |
| 10 | ADD DR GR | 56 |
| 11 | WRM DR | 37 |
| 12 | LLS 2 | 72 |
| 13 | LMS 9 | 89 |
| 14 | CPR AR GR | 42 |
| 15 | RDM DR | 27 |
| 16 | LLS 2 | 72 |
| 17 | LMS 8 | 88 |
| 18 | CPR AR GR | 42 |
| 19 | RDM GR | 2B |
| 1A | SUB DR GR | 66 |
| 1B | WRM DR | 37 |
| 1C | LLS 0 | 70 |
| 1D | LMS F | 8F |
| 1E | CPR AR GR | 42 |
| 1F | CFR | 9F |
| 20 | WRM GR | 3B |
| 21 | LLS 0 | 70 |
| 22 | LMS 0 | 80 |
| 23 | CPR AR GR | 42 |
| 24 | JMP | 1F |
|  |  |  |
|  |  |  |

## Table IV.2: Initial Test Data Stored atMemory Addresses (in Hex)

|  |  |  |  |
| --- | --- | --- | --- |
| **Memory Address** | **Data** | **Memory Address** | **Data** |
| 80 | 15 | 92 | 44 |
| 81 | 35 | 82 | 12 |
| 91 | 35 |  |  |

## IV.2 Test Results

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; Dec 7 22:12 2014

0 mem[80]=15 mem[81]=35 mem[82]=12

510 mem[80]=5a mem[81]=35 mem[82]=12

1110 mem[80]=5a mem[81]=6a mem[82]=12

1710 mem[80]=5a mem[81]=6a mem[82]=32

2430 mem[80]=9f mem[81]=6a mem[82]=32

3030 mem[80]=9f mem[81]=9f mem[82]=32

3630 mem[80]=9f mem[81]=9f mem[82]=12

$finish called from file "/home/wa/wali9257/ee271/s8sp/implementation/rtl\_vcs/../../s8sp\_tb/tb\_s8sp.v", line 128.

$finish at simulation time 4000000

V C S S i m u l a t i o n R e p o r t

Time: 4000000 ps

CPU Time: 0.140 seconds; Data structure size: 0.0Mb

Sun Dec 7 22:12:54 2014

# V. Synthesis and Optimizations of Whole Design (processor)

## Table V.1: Synthesis Constraints and Results

**Table V.II**: Variation in Area and Power with the change in clock period (Constant Area)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trial**  **No**. | **Constraint settings** | | **Synthesis results** | | |
| **MAX\_AREA** | **CLOCK(ns)** | **SLACK** | **AREA** | **POWER(mW)** |
| **1** | 500 | 5.0 | 0.05 | 1122.0 | 1.9644 |
| **2** | 500 | 4.0 | 0.05 | 1126.0 | 2.9748 |
| **3** | 500 | 3.0 | 0.02 | 1108.5 | 3.9498 |
| **4** | 500 | 2.0 | 0.02 | 1165.0 | 5.9353 |
| **5** | 500 | 1.8 | 0.01 | 1179.0 | 6.5928 |
| **6** | 500 | 1.7 | 0.00 | 1179.5 | 9.4142 |

**Table V.II**: Variation in Area and Power with the change in Area (Constant clock period)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Trial**  **No**. | **Constraint settings** | | **Synthesis results** | | |
| **AREA** | **CLOCK(ns)** | **SLACK** | **AREA(sq.nm)** | **POWER(mW)** |
| **1** | 300 | 2.0 | 0.02 | 1165 | 5.9353 |
| **2** | 400 | 2.0 | 0.02 | 1165 | 5.9353 |
| **3** | 1000 | 2.0 | 0.19 | 1165 | 5.9353 |
| **4** | 2000 | 2.0 | 0.13 | 1203 | 6.0221 |
| **5** | 3000 | 2.0 | 0.19 | 1165 | 5.9353 |

**Figure V.1**: Power versus Clock of the Final Design

**Figure V.1**: Area versus Clock of the Final Design

## V.III Relationship between Area Power and operating clock frequency

Various synthesis runs were carried out to obtain a better tradeoff between area, power and timing. I analyzed the total area consumed and total power consumed by the design for different values of the operating clock frequency. From Table V.I and V.II we can see that as the operating clock frequency increases area and power consumption of the design increases. For my design the best operating clock frequency obtained is 588 MHz (1.7ns).

# VI. Gate-Level Simulations/Tests of Whole Design (processor)

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; Dec 7 23:16 2014

VCD+ Writer I-2014.03-2 Copyright (c) 1991-2014 by Synopsys Inc.

The file '/home/wa/wali9257/ee271/s8sp/implementation/netlist\_vcs/inter.vpd' was opened successfully.

dve> run

0 mem[80]=15 mem[81]=25 mem[82]=54

198 mem[80]=zZ mem[81]=25 mem[82]=54

198 mem[80]=4a mem[81]=25 mem[82]=54

438 mem[80]=4a mem[81]=zZ mem[82]=54

438 mem[80]=4a mem[81]=5a mem[82]=54

678 mem[80]=4a mem[81]=5a mem[82]=zZ

678 mem[80]=4a mem[81]=5a mem[82]=f0

1086 mem[80]=zZ mem[81]=5a mem[82]=f0

1086 mem[80]=8f mem[81]=5a mem[82]=f0

1326 mem[80]=8f mem[81]=zZ mem[82]=f0

1326 mem[80]=8f mem[81]=8f mem[82]=f0

1566 mem[80]=8f mem[81]=8f mem[82]=zZ

1566 mem[80]=8f mem[81]=8f mem[82]=54

1974 mem[80]=zZ mem[81]=8f mem[82]=54

1974 mem[80]=d4 mem[81]=8f mem[82]=54

$finish called from file "tb\_s8sp.v", line 128.

$finish at simulation time 2000000

Simulation complete, time is 2000000 ps.

V C S S i m u l a t i o n R e p o r t

Time: 2000000 ps

CPU Time: 0.040 seconds; Data structure size: 0.2Mb

Sun Dec 7 23:17:17 2014

# VII. Conclusion

Successfully designed and implemented a simple 8 bit scalar processor. To obtain a faster response for arithmetic instructions a Carry look ahead adder circuit was designed and tested independently and then integrated into the processor design. In order for effective utilization of resources the same carry look ahead adder circuit was used for subtraction with some additional combinational logic. Synthesis trails were carried out to obtain the better tradeoff between area power and speed. Maximum operating clock frequency of the processor obtained is 588Mhz. From synthesis trails it was known that as operating clock frequency increases power consumption and area utilized by the circuit increases.

# VIII. References

* Dr. Thuy Le, EE271 Fall 2014 class notes.
* Dr. Thuy Le, project handout material.
* Synopsys DC Synthesis tutorial.
* Verilog HDL: A Guide to Digital Design and Synthesis, Second Edition - Samir Palnitkar
* Advanced Digital Design with the Verilog HDL (2nd Edition)- Michael ciletti

# Appendix A

**A.1** Scripts and/or Commands Used for Simulation and Synthesis

### A.1.1 RTL simulation command

vcs -RPP /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/addr\_reg.v /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/data\_reg.v /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/flag\_reg.v /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/gen\_reg.v /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/local\_ir\_reg.v /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/program\_cnt\_reg.v /home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/alu\_block.v /home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/CLA8bit.v /home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/CLA\_N\_bit.v /home/wa/wali9257/ee271/s8sp/data\_block/data\_block\_top/data\_block.v /home/wa/wali9257/ee271/s8sp/ctrl\_block/sys\_controller.v /home/wa/wali9257/ee271/s8sp/s8sp\_top/s8sp.v /home/wa/wali9257/ee271/s8sp/implementation/rtl\_vcs/../../s8sp\_tb/tb\_s8sp.v

### 

### A.2.1 Synthesis\_script

dc\_shell -xg -f synthesis.script

set link\_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_BCCOM25 /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_WCCOM25 }

set target\_library {/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_BCCOM25 /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_WCCOM25 }

set synthetic\_library {dw\_foundation.sldb standard.sldb}

set\_min\_library /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_WCCOM25 -min\_version /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_BCCOM25

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/addr\_reg.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/data\_reg.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/flag\_reg.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/gen\_reg.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/local\_ir\_reg.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/program\_cnt\_reg.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/CLA\_N\_bit.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/CLA8bit.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/alu\_block.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/data\_block/data\_block\_top/data\_block.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/ctrl\_block/sys\_controller.v

analyze -f verilog /home/wa/wali9257/ee271/s8sp/s8sp\_top/s8sp.v

elaborate s8sp

current\_design s8sp

uniquify -reference CLA\_N\_bit

link

#set\_flatten -design s8sp

#set\_flatten -design alu\_block

#ungroup -all

check\_design > reports/design\_report

#create\_clock clk -name clk -period 2.800000

create\_clock clk -name clk -period 2

set\_propagated\_clock clk

set\_clock\_uncertainty .25 clk

set\_max\_delay 1 -from [all\_inputs]

set\_max\_delay 1 -to [all\_outputs]

#set\_max\_delay 2 -to sum

#set\_load 160 sum

set\_max\_area 1000

#compile -map\_effort high -ungroup\_all

compile -map\_effort high

report\_cell > reports/cell\_report

report\_net > reports/net\_report

update\_timing

report\_timing -max\_paths 10 -group clk > reports/timing\_report

report\_area > reports/area\_report

report\_power > reports/power\_report

write -hierarchy -format verilog -output reports/s8sp\_netlist.v

#write -format verilog -output reports/s8sp\_netlist.v

quit

Appendix B

# Completed Verilog Source Codes and Test benches

**Top Level Module of the Design**

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 02:05:08 10/28/2014

// Design Name: Simple 8 bit Scalar Processor

// Module Name: s8sp

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module s8sp(clk,reset,add,dat,wrt,rd );

input clk;

input reset;

output[7:0] add;

inout [7:0] dat;

output wrt;

output rd;

//For AR Connections from Controller to Data\_block

wire load\_ar;

wire ar\_on\_addr;

wire ar\_2\_data;

//For DR Connections from Controller to Data\_block

wire load\_dr;

wire dr\_2\_data;

//For GR Connections from Controller to Data\_block

wire load\_lsb\_gr;

wire load\_msb\_gr;

wire gr\_2\_data;

//For PR Connections from Controller to Data\_block

wire load\_ar\_2\_pr;

wire inc\_pr;

wire pr\_2\_data;

wire pr\_on\_addr;

//For IR Connections from Controller to Data\_block

wire load\_ir;

wire ir\_2\_data;

wire [7:0] ir\_code;

//For ALU Connections from Controller to Data\_block

wire sub\_nadd;

wire alu\_2\_data;

wire [1:0] add\_oprnd1\_sel;

wire [1:0] add\_oprnd2\_sel;

//For Flag connections

wire flag\_2\_data;

data\_block u1\_data\_block\_instantiation (

//System Signals

.clk (clk),

.reset (reset),

//System Bus

.data\_bus (dat),

.addr\_bus (add),

//AR related signals

.load\_ar(load\_ar),

.ar\_on\_addr(ar\_on\_addr),

.ar\_2\_data(ar\_2\_data),

//DR related signals

.load\_dr(load\_dr),

.dr\_2\_data(dr\_2\_data),

//GR related signals

.load\_lsb\_gr(load\_lsb\_gr),

.load\_msb\_gr(load\_msb\_gr),

.gr\_2\_data(gr\_2\_data),

//PR related signals

.load\_ar\_2\_pr(load\_ar\_2\_pr),

.inc\_pr(inc\_pr),

.pr\_2\_data(pr\_2\_data),

.pr\_on\_addr(pr\_on\_addr),

//Internal IR related signals

.load\_ir(load\_ir),

.ir\_2\_data(ir\_2\_data),

.ir\_code(ir\_code), //used by controller for decoding instructions

//ALU related signals

.sub\_nadd(sub\_nadd),

.add\_oprnd1\_sel(add\_oprnd1\_sel),

.add\_oprnd2\_sel(add\_oprnd2\_sel),

.alu\_2\_data (alu\_2\_data),

//FLAG related signals

.flag\_2\_data (flag\_2\_data)

);

//--------------------------------------------------------------

sys\_controller u1\_sys\_controller\_inst (

//System related signals

.clk (clk),

.reset (reset),

.rd\_mem (rd),

.wr\_mem (wrt),

//AR related signals

.ctrl\_load\_ar(load\_ar),

.ctrl\_ar\_on\_addr(ar\_on\_addr),

.ctrl\_ar\_2\_data(ar\_2\_data),

//DR related signals

.ctrl\_load\_dr(load\_dr),

.ctrl\_dr\_2\_data(dr\_2\_data),

//GR related signals

.ctrl\_load\_lsb\_gr(load\_lsb\_gr),

.ctrl\_load\_msb\_gr(load\_msb\_gr),

.ctrl\_gr\_2\_data(gr\_2\_data),

//PR related signals

.ctrl\_load\_ar\_2\_pr(load\_ar\_2\_pr),

.ctrl\_inc\_pr(inc\_pr),

.ctrl\_pr\_2\_data(pr\_2\_data),

.ctrl\_pr\_on\_addr(pr\_on\_addr),

//Internal IR related signals

.ctrl\_load\_ir(load\_ir),

.ctrl\_ir\_2\_data(ir\_2\_data),

.ctrl\_ir\_code(ir\_code),

//ALU related signals

.ctrl\_alu\_2\_data(alu\_2\_data),

.ctrl\_sub\_nadd(sub\_nadd),

.ctrl\_add\_oprnd1\_sel(add\_oprnd1\_sel),

.ctrl\_add\_oprnd2\_sel(add\_oprnd2\_sel),

//FLAG related signals

.ctrl\_flag\_2\_data (flag\_2\_data)

);

endmodule

**Controller module of the design**

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 21:49:57 10/27/2014

// Design Name: System Controller

// Module Name: sys\_controller

// Project Name: S8SP

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module sys\_controller( clk,reset,rd\_mem,wr\_mem,ctrl\_load\_ar,ctrl\_ar\_on\_addr,ctrl\_ar\_2\_data,ctrl\_load\_dr,ctrl\_dr\_2\_data,

ctrl\_load\_lsb\_gr,ctrl\_load\_msb\_gr,ctrl\_gr\_2\_data,ctrl\_load\_ar\_2\_pr,ctrl\_inc\_pr,ctrl\_pr\_2\_data,ctrl\_pr\_on\_addr,ctrl\_load\_ir,

ctrl\_ir\_2\_data,ctrl\_ir\_code,ctrl\_alu\_2\_data,ctrl\_sub\_nadd,ctrl\_add\_oprnd1\_sel,ctrl\_add\_oprnd2\_sel,ctrl\_flag\_2\_data );

//System related signals

input clk;

input reset;

output reg rd\_mem;

output reg wr\_mem;

//AR related signals

output reg ctrl\_load\_ar;

output reg ctrl\_ar\_on\_addr;

output reg ctrl\_ar\_2\_data;

//DR related signals

output reg ctrl\_load\_dr;

output reg ctrl\_dr\_2\_data;

//GR related signals

output reg ctrl\_load\_lsb\_gr;

output reg ctrl\_load\_msb\_gr;

output reg ctrl\_gr\_2\_data;

//PR related signals

output reg ctrl\_load\_ar\_2\_pr;

output reg ctrl\_inc\_pr;

output reg ctrl\_pr\_2\_data;

output reg ctrl\_pr\_on\_addr;

//Internal IR related signals

output reg ctrl\_load\_ir;

output reg ctrl\_ir\_2\_data;

input [7:0] ctrl\_ir\_code;

//ALU related signals

output reg ctrl\_alu\_2\_data;

output reg ctrl\_sub\_nadd;

output reg [1:0] ctrl\_add\_oprnd1\_sel;

output reg [1:0] ctrl\_add\_oprnd2\_sel;

//FLAG related signals

output reg ctrl\_flag\_2\_data;

//State Assignment

parameter RST = 2'b00;

parameter FETCH = 2'b01;

parameter DECODE = 2'b10;

parameter EXECUTE = 2'b11;

//Register Assignment

parameter AR = 2'b00;

parameter DR = 2'b01;

parameter GR = 2'b10;

parameter PR = 2'b11;

//Instruction opcodes

parameter NOP = 4'b0000;

parameter JMP = 4'b0001;

parameter RDM = 4'b0010;

parameter WRM = 4'b0011;

parameter CPR = 4'b0100;

parameter ADD = 4'b0101;

parameter SUB = 4'b0110;

parameter LLS = 4'b0111;

parameter LMS = 4'b1000;

parameter CFR = 4'b1001;

reg [1:0] present\_state;

reg [1:0] next\_state;

always @(posedge clk)

begin

if(reset)

present\_state <= 2'b00;

else

present\_state <= next\_state;

end

always @(present\_state or reset or ctrl\_ir\_code)

begin

rd\_mem = 0;

wr\_mem = 0;

ctrl\_load\_ar = 0;

ctrl\_ar\_on\_addr = 0;

ctrl\_ar\_2\_data = 0;

ctrl\_load\_dr = 0;

ctrl\_dr\_2\_data = 0;

ctrl\_load\_lsb\_gr = 0;

ctrl\_load\_msb\_gr = 0;

ctrl\_gr\_2\_data = 0;

ctrl\_load\_ar\_2\_pr = 0;

ctrl\_inc\_pr = 0;

ctrl\_pr\_2\_data = 0;

ctrl\_pr\_on\_addr = 0;

ctrl\_load\_ir = 0;

ctrl\_ir\_2\_data = 0;

//ctrl\_ir\_code,

ctrl\_alu\_2\_data = 0;

ctrl\_sub\_nadd = 0;

ctrl\_add\_oprnd1\_sel = 2'b01;

ctrl\_add\_oprnd2\_sel = 2'b10;

ctrl\_flag\_2\_data = 0;

case (present\_state)

RST : begin

next\_state = reset? RST : FETCH;

end

FETCH : begin

next\_state = DECODE;

rd\_mem = 1;

ctrl\_pr\_on\_addr = 1;

ctrl\_load\_ir = 1;

ctrl\_inc\_pr = 1;

end

DECODE : begin

next\_state = EXECUTE;

end

EXECUTE : begin

next\_state = FETCH;

case (ctrl\_ir\_code [7:4]) //synopsys full\_case

NOP : ;

JMP : begin

ctrl\_load\_ar\_2\_pr = 1;

ctrl\_ar\_2\_data = 1;

end

RDM : begin

rd\_mem = 1;

ctrl\_ar\_on\_addr = 1;

case (ctrl\_ir\_code [3:2])

AR : ctrl\_load\_ar = 1;

DR : ctrl\_load\_dr = 1;

GR : begin ctrl\_load\_lsb\_gr = 1;

ctrl\_load\_msb\_gr = 1;

end

PR : ctrl\_load\_ar\_2\_pr = 1;

endcase

end

WRM : begin

wr\_mem = 1;

ctrl\_ar\_on\_addr = 1;

case (ctrl\_ir\_code [3:2])

AR : ctrl\_ar\_2\_data = 1;

DR : ctrl\_dr\_2\_data = 1;

GR : ctrl\_gr\_2\_data = 1;

PR : ctrl\_pr\_2\_data = 1;

endcase

end

CPR : begin

case (ctrl\_ir\_code [3:0])

4'b0000 : ;

4'b0001 : begin ctrl\_dr\_2\_data = 1; ctrl\_load\_ar = 1; end

4'b0010 : begin ctrl\_gr\_2\_data = 1; ctrl\_load\_ar = 1; end

4'b0011 : begin ctrl\_pr\_2\_data = 1; ctrl\_load\_ar = 1; end

//---------------------------------------------------------

4'b0100 : begin ctrl\_ar\_2\_data = 1; ctrl\_load\_dr = 1; end

4'b0101 : ;

4'b0110 : begin ctrl\_gr\_2\_data = 1; ctrl\_load\_dr = 1; end

4'b0111 : begin ctrl\_pr\_2\_data = 1; ctrl\_load\_dr = 1; end

//----------------------------------------------------------

//-----------------------------------------------------------

4'b1000 : begin ctrl\_ar\_2\_data = 1; ctrl\_load\_lsb\_gr = 1; ctrl\_load\_msb\_gr = 1; end

4'b1001 : begin ctrl\_dr\_2\_data = 1; ctrl\_load\_lsb\_gr = 1; ctrl\_load\_msb\_gr = 1; end

4'b1010 : ;

4'b1011 : begin ctrl\_pr\_2\_data = 1; ctrl\_load\_lsb\_gr = 1; ctrl\_load\_msb\_gr = 1; end

//----------------------------------------------------------------

//----------------------------------------------------------------

4'b1100 : begin ctrl\_ar\_2\_data = 1; ctrl\_load\_ar\_2\_pr = 1; end

4'b1101 : begin ctrl\_dr\_2\_data = 1; ctrl\_load\_ar\_2\_pr = 1; end

4'b1110 : begin ctrl\_gr\_2\_data = 1; ctrl\_load\_ar\_2\_pr = 1; end

4'b1111 : ;

endcase

end

ADD : begin ctrl\_sub\_nadd = 0;

ctrl\_add\_oprnd1\_sel = ctrl\_ir\_code [3:2];

ctrl\_add\_oprnd2\_sel = ctrl\_ir\_code [1:0];

ctrl\_alu\_2\_data = 1;

case (ctrl\_ir\_code [3:2])

AR : ctrl\_load\_ar = 1;

DR : ctrl\_load\_dr = 1;

GR : begin ctrl\_load\_lsb\_gr = 1;

ctrl\_load\_msb\_gr = 1;

end

PR : ctrl\_load\_ar\_2\_pr = 1;

endcase

end

SUB : begin ctrl\_sub\_nadd = 1;

ctrl\_add\_oprnd1\_sel = ctrl\_ir\_code [3:2];

ctrl\_add\_oprnd2\_sel = ctrl\_ir\_code [1:0];

ctrl\_alu\_2\_data = 1;

case (ctrl\_ir\_code [3:2])

AR : ctrl\_load\_ar = 1;

DR : ctrl\_load\_dr = 1;

GR : begin ctrl\_load\_lsb\_gr = 1;

ctrl\_load\_msb\_gr = 1;

end

PR : ctrl\_load\_ar\_2\_pr = 1;

endcase

end

LLS : begin ctrl\_ir\_2\_data = 1; ctrl\_load\_lsb\_gr = 1; end

LMS : begin ctrl\_ir\_2\_data = 1; ctrl\_load\_msb\_gr = 1; end

CFR : begin ctrl\_flag\_2\_data = 1;

ctrl\_load\_lsb\_gr = 1; end

default : next\_state = RST ;

endcase //endcase of EXECUTE block

end //end of EXECUTE begin block

endcase

end

endmodule

---------------------------------------------------------------------------------------------------------------------

**Data Handling Block**

`timescale 1ns / 1ps

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 13:57:22 10/26/2014

// Design Name:

// Module Name: data\_block

// Project Name: S8SP

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module data\_block(clk,reset,data\_bus,addr\_bus,load\_ar,ar\_on\_addr,ar\_2\_data,load\_dr,dr\_2\_data,

load\_lsb\_gr,load\_msb\_gr,gr\_2\_data,load\_ar\_2\_pr,inc\_pr,pr\_2\_data,pr\_on\_addr,load\_ir,ir\_2\_data,

ir\_code,sub\_nadd,add\_oprnd1\_sel,add\_oprnd2\_sel,alu\_2\_data,flag\_2\_data);

//System Bus

inout [7:0] data\_bus;

output [7:0] addr\_bus;

//System Signals

input clk;

input reset;

//AR related signals

input load\_ar;

input ar\_on\_addr;

input ar\_2\_data;

//DR related signals

input load\_dr;

input dr\_2\_data;

//GR related signals

input load\_lsb\_gr;

input load\_msb\_gr;

input gr\_2\_data;

//PR related signals

input load\_ar\_2\_pr;

input inc\_pr;

input pr\_2\_data;

input pr\_on\_addr;

//Internal IR related signals

input load\_ir;

input ir\_2\_data;

output [7:0] ir\_code; //used by controller for decoding instructions

//ALU related signals

input sub\_nadd;

input [1:0] add\_oprnd1\_sel;

input [1:0] add\_oprnd2\_sel;

input alu\_2\_data;

//Flag\_reg related signals

input flag\_2\_data;

wire [7:0] pr\_on\_bus;

wire [7:0] dr\_on\_data;

wire [7:0] gr\_on\_data;

wire [7:0] ar\_on\_bus;

wire [8:0] alu\_out;

wire [7:0] ir\_out;

wire [3:0] flag\_on\_data;

reg [7:0] mux\_out1;

reg [7:0] mux\_out2;

program\_cnt\_reg u1\_pr\_instance (

.clk (clk),

.reset (reset),

.inc\_pr (inc\_pr),

.pr\_on\_bus (pr\_on\_bus),

.load\_ar\_2\_pr (load\_ar\_2\_pr),

.data\_on\_pr (data\_bus)

);

data\_reg u1\_dr\_instance (

.clk (clk),

.reset (reset),

.data\_on\_dr (data\_bus),

.dr\_on\_data (dr\_on\_data),

.load\_dr (load\_dr)

);

gen\_reg u1\_gr\_instance (

.clk (clk),

.reset (reset),

.load\_lsb\_gr (load\_lsb\_gr),

.load\_msb\_gr (load\_msb\_gr),

.data\_on\_gr (data\_bus),

.gr\_on\_data (gr\_on\_data)

);

addr\_reg u1\_ar\_instance (

.clk (clk),

.reset (reset),

.data\_on\_ar (data\_bus),

.ar\_on\_bus (ar\_on\_bus),

.load\_ar (load\_ar)

);

local\_ir\_reg u1\_ir\_instance (

.clk (clk),

.reset (reset),

.load\_ir (load\_ir),

.ir\_out (ir\_out),

.data\_on\_ir (data\_bus)

);

alu\_block u1\_alu\_instance (

.reg\_in1 (mux\_out1),

.reg\_in2 (mux\_out2),

//.clk (clk),

//.reset (reset),

.sub\_nadd (sub\_nadd),

.alu\_out (alu\_out)

);

flag\_reg u1\_flag\_reg\_instance (

.clk (clk),

.reset (reset),

.alu\_2\_data (alu\_2\_data),

.alu\_data (alu\_out),

.flag\_reg (flag\_on\_data)

);

always @(\*)

begin : mux\_block1

case (add\_oprnd1\_sel)

2'b00 : mux\_out1 = ar\_on\_bus;

2'b01 : mux\_out1 = dr\_on\_data;

2'b10 : mux\_out1 = gr\_on\_data;

2'b11 : mux\_out1 = pr\_on\_bus;

endcase

end

always @(\*)

begin : mux\_block2

case (add\_oprnd2\_sel)

2'b00 : mux\_out2 = ar\_on\_bus;

2'b01 : mux\_out2 = dr\_on\_data;

2'b10 : mux\_out2 = gr\_on\_data;

2'b11 : mux\_out2 = pr\_on\_bus;

endcase

end

assign addr\_bus = ar\_on\_addr ? ar\_on\_bus : 8'bzzzz\_zzzz;

assign addr\_bus = pr\_on\_addr ? pr\_on\_bus : 8'bzzzz\_zzzz;

assign data\_bus = pr\_2\_data ? pr\_on\_bus : 8'bzzzz\_zzzz;

assign data\_bus = dr\_2\_data ? dr\_on\_data : 8'bzzzz\_zzzz;

assign data\_bus = gr\_2\_data ? gr\_on\_data : 8'bzzzz\_zzzz;

assign data\_bus = ar\_2\_data ? ar\_on\_bus : 8'bzzzz\_zzzz;

assign data\_bus = alu\_2\_data ? alu\_out[7:0] : 8'bzzzz\_zzzz;

assign data\_bus = flag\_2\_data ?{4'b0000,flag\_on\_data} : 8'bzzzz\_zzzz;

assign data\_bus = ir\_2\_data ? ir\_out : 8'bzzzz\_zzzz;

assign ir\_code = ir\_out;

endmodule

---------------------------------------------------------------------------------------------------------------------

**ALU Block**

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 14:22:43 10/27/2014

// Design Name:

// Module Name: alu\_block

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments: This block needs some more enhancemt. Need to add 4 flag register

//

//////////////////////////////////////////////////////////////////////////////////

module alu\_block(reg\_in1,reg\_in2,sub\_nadd,alu\_out);

input [7:0] reg\_in1;

input [7:0] reg\_in2;

input sub\_nadd;

output [8:0] alu\_out;

//reg [8:0] alu\_out;

wire [7:0] xor\_out;

genvar i;

generate

for (i=0; i<8; i=i+1)

begin : SUB\_BLOCK

xor xor\_u1 (xor\_out[i], reg\_in2[i], sub\_nadd);

end

endgenerate

CLA8bit cla8bit\_u1 (.in1(reg\_in1), .in2(xor\_out), .sum(alu\_out[7:0]), .cin(sub\_nadd), .cout(alu\_out[8]));

endmodule

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 21:41:48 11/07/2014

// Design Name:

// Module Name: CLA8bit

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module CLA8bit (in1,in2,sum,cin,cout);

input [7:0] in1,in2;

input cin;

output [7:0] sum;

output cout;

wire cnxt;

CLA\_N\_bit #4 cla\_inst0 (.in1(in1[3:0]),.in2(in2[3:0]),.sum(sum[3:0]),.cin(cin),.cout(cnxt));

CLA\_N\_bit #4 cla\_inst2 (.in1(in1[7:4]),.in2(in2[7:4]),.sum(sum[7:4]),.cin(cnxt),.cout(cout));

endmodule

module CLA\_N\_bit (in1,in2,sum,cin,cout);

parameter N=4;

input [N-1 :0 ] in1,in2;

input cin;

output [N-1 : 0] sum;

output cout;

reg [N : 0] c;

integer i;

always @(\*)

begin

c[0] = cin;

for (i=0; i<N; i=i+1) begin:CARRY\_BLK

c[i+1] = (in1[i] & in2[i]) | ((in1[i] ^ in2[i]) & c[i]);

end

end

genvar j;

generate

for( j=0; j<N; j=j+1) begin:SUM\_BLK

xor u1 (sum[j],c[j],in1[j],in2[j]);

end

endgenerate

assign cout = c[N];

endmodule

-----------------------------------------------------------------------------------------------------------------

**Register\_Modules**

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 00:18:50 10/27/2014

// Design Name: Address\_Register Block

// Module Name: addr\_reg

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module addr\_reg(clk,reset,data\_on\_ar,ar\_on\_bus,load\_ar);

input clk;

input reset;

input load\_ar;

input [7:0] data\_on\_ar;

output [7:0] ar\_on\_bus;

reg [7:0] ar\_on\_bus;

always @(posedge clk)

begin

if(reset)

ar\_on\_bus <= 8'd0;

else

begin

if(load\_ar)

ar\_on\_bus <= data\_on\_ar;

end

end

endmodule

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 21:18:20 10/26/2014

// Design Name:

// Module Name: data\_reg

// Project Name: S8SP

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module data\_reg(clk,reset,data\_on\_dr,dr\_on\_data,load\_dr);

input clk;

input reset;

input load\_dr;

input [7:0] data\_on\_dr;

output [7:0] dr\_on\_data;

reg [7:0] dr\_on\_data;

always @(posedge clk)

begin: DR\_Block

if(reset)

dr\_on\_data <= 8'd0;

else

begin

if(load\_dr)

dr\_on\_data <= data\_on\_dr;

end

end

endmodule

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 13:43:48 11/08/2014

// Design Name:

// Module Name: flag\_reg

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module flag\_reg(clk,reset,alu\_2\_data,alu\_data,flag\_reg);

input clk;

input reset;

input alu\_2\_data;

input [8:0] alu\_data;

output [3:0] flag\_reg;

reg [3:0] flag\_reg;

always @(posedge clk)

begin

if(reset)

flag\_reg <= 4'b0000;

else if(alu\_2\_data)

begin

flag\_reg[3] <= ~|alu\_data;

flag\_reg[2] <= (alu\_data[7]^alu\_data[8]);

flag\_reg[1] <= alu\_data[8];

flag\_reg[0] <= alu\_data[7]^alu\_data[8];

end

else

flag\_reg <= flag\_reg;

end

endmodule

//////////////////////////////////////////////////////////////////////////////////

// Company: San Jose State University

// Engineer:

//

// Create Date: 21:33:15 10/26/2014

// Design Name: Genral Register Block

// Module Name: gen\_reg

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module gen\_reg(clk,reset,load\_lsb\_gr,load\_msb\_gr,data\_on\_gr,gr\_on\_data );

input clk;

input reset;

input load\_lsb\_gr;

input load\_msb\_gr;

input [7:0] data\_on\_gr;

output [7:0] gr\_on\_data;

reg [7:0] gr\_on\_data;

always @(posedge clk)

begin : Genral\_Register\_Block

if( reset)

gr\_on\_data <= 8'd0;

else

begin

if(load\_lsb\_gr && !load\_msb\_gr)

gr\_on\_data[3:0] <= data\_on\_gr[3:0];

else if(load\_msb\_gr && !load\_lsb\_gr)

gr\_on\_data[7:4] <= data\_on\_gr[3:0];

else if(load\_msb\_gr && load\_lsb\_gr)

gr\_on\_data <= data\_on\_gr;

end

end

endmodule

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 00:34:33 10/27/2014

// Design Name:

// Module Name: local\_ir\_reg

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module local\_ir\_reg(clk,reset,load\_ir,ir\_out,data\_on\_ir);

input clk;

input reset;

input load\_ir;

input [7:0] data\_on\_ir;

output [7:0] ir\_out;

reg [7:0] ir\_out;

always @(posedge clk)

begin

if(reset)

ir\_out <= 8'd0;

else

begin

if(load\_ir)

ir\_out <= data\_on\_ir;

end

end

endmodule

//////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 14:45:15 10/26/2014

// Design Name:

// Module Name: program\_cnt\_reg

// Project Name:

// Target Devices:

// Tool versions:

// Description:

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

//////////////////////////////////////////////////////////////////////////////////

module program\_cnt\_reg(clk,reset,inc\_pr,pr\_on\_bus,load\_ar\_2\_pr,data\_on\_pr);

input clk;

input reset;

input inc\_pr;

input load\_ar\_2\_pr;

input [7:0] data\_on\_pr;

output [7:0] pr\_on\_bus;

reg [7:0] pr\_on\_bus;

always @(posedge clk)

begin : PR\_Block

if (reset)

pr\_on\_bus <= 8'd0;

else

begin

if(load\_ar\_2\_pr)

pr\_on\_bus <= data\_on\_pr;

else if( inc\_pr)

pr\_on\_bus <= pr\_on\_bus+1;

end

end

endmodule

**---------------------------------------------------------------------------------------------------------------------**

**TestBench Used to test the design module**

**`timescale 1ns / 1ps**

////////////////////////////////////////////////////////////////////////////////

// Company:

// Engineer:

//

// Create Date: 03:28:58 11/03/2014

// Design Name: s8sp

// Module Name: D:/xilinx\_location/S8SP/tb\_s8sp.v

// Project Name: S8SP

// Target Device:

// Tool versions:

// Description:

//

// Verilog Test Fixture created by ISE for module: s8sp

//

// Dependencies:

//

// Revision:

// Revision 0.01 - File Created

// Additional Comments:

//

////////////////////////////////////////////////////////////////////////////////

`define clk\_period\_by2 10

module tb\_s8sp;

// Inputs

reg clk;

reg reset;

// Outputs

wire [7:0] add;

wire wrt;

wire rd;

// Bidirs

wire [7:0] dat;

reg [7:0] out\_data;

//wire [7:0] out\_data;

reg [7:0] mem [0:255];

// Instantiate the Unit Under Test (UUT)

s8sp uut (

.clk(clk),

.reset(reset),

.add(add),

.dat(dat),

.wrt(wrt),

.rd(rd)

);

//Memory Initialisation

initial

begin

$readmemh ( "mem.txt", mem, 8'h00, 8'hff );

end

initial begin

// Initialize Inputs

clk = 0;

reset = 1;

end

always # `clk\_period\_by2 clk = ~clk;

initial begin

#40 reset = 0;

end

// Read memory block

always @(rd or add)

begin

if(rd)

out\_data = mem[add];

end

//Write memory block

always @(wrt or add or dat)

begin

if( wrt )

mem[add] = dat;

end

assign dat = rd ? out\_data : 8'bzzzz\_zzzz;

initial begin

$dumpfile("s8sp\_wave.vcd");

$dumpvars(0,uut);

end

initial #4000 $finish;

initial

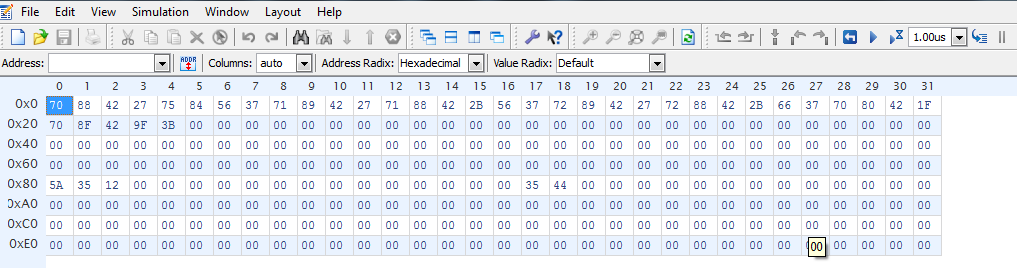
begin

$monitor("%d \t mem[80]=%h \t mem[81]=%h \t mem[82]=%h \t", $time,mem[128],mem[129],mem[130]);

end

endmodule

**Mem.txt file is as below**

****

Appendix C

# Reports and Circuits from EDA Tools

## C.1 Reports (contents) from RTL (Pre-synthesis) Simulations (VCS or NCVERILOG)

Note-[VIRSIM\_GONE] Virsim not supported

Virsim is no longer supported as warned in earlier VCS releases.

DVE has replaced Virsim as the default Debug GUI since VCS 2006.06 release.

The Virsim vcs switches -RI, -RIG, and -RPP are also no longer supported.

Please invoke DVE as follows:

Interactive Mode compile time: >vcs <other options> -R -gui

Interactive Mode run time: >simv <other optoins> -gui

Post Process Mode >dve &

Contact Synopsys VCS Support for more details: vcs\_support@synopsys.com .

Chronologic VCS (TM)

Version I-2014.03-2 -- Sun Dec 7 22:12:04 2014

Copyright (c) 1991-2014 by Synopsys Inc.

ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys Inc.

and may be used and disclosed only as authorized in a license agreement

controlling such use and disclosure.

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/addr\_reg.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/data\_reg.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/flag\_reg.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/gen\_reg.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/local\_ir\_reg.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/reg\_blocks/program\_cnt\_reg.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/alu\_block.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/CLA8bit.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/alu\_block/CLA\_N\_bit.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/data\_block/data\_block\_top/data\_block.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/ctrl\_block/sys\_controller.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/s8sp\_top/s8sp.v'

Parsing design file '/home/wa/wali9257/ee271/s8sp/implementation/rtl\_vcs/../../s8sp\_tb/tb\_s8sp.v'

Top Level Modules:

tb\_s8sp

TimeScale is 1 ns / 1 ps

Starting vcs inline pass...

2 modules and 0 UDP read.

However, due to incremental compilation, no re-compilation is necessary.

rm -f \_csrc\*.so linux\_scvhdl\_\*.so pre\_vcsobj\_\*.so share\_vcsobj\_\*.so

ld -m elf\_i386 -shared -o .//../simv.daidir//\_csrc0.so SIM\_l.o 5NrI\_d.o 5NrIB\_d.o

ld -m elf\_i386 -shared -o .//../simv.daidir//pre\_vcsobj\_1\_1.so --whole-archive pre\_vcsobj\_1\_1.a --no-whole-archive

if [ -x ../simv ]; then chmod -x ../simv; fi

g++ -o ../simv -m32 -m32 -Wl,-rpath-link=./ -Wl,-rpath='$ORIGIN'/simv.daidir/ -Wl,-rpath='$ORIGIN'/simv.daidir//scsim.db.dir \_csrc0.so pre\_vcsobj\_1\_1.so rmapats\_mop.o rmapats.o rmar.o /apps/synopsys/VCSMX\_NEW/linux/lib/libzerosoft\_rt\_stubs.so /apps/synopsys/VCSMX\_NEW/linux/lib/libvirsim.so /apps/synopsys/VCSMX\_NEW/linux/lib/librterrorinf.so /apps/synopsys/VCSMX\_NEW/linux/lib/libsnpsmalloc.so /apps/synopsys/VCSMX\_NEW/linux/lib/libvcsnew.so /apps/synopsys/VCSMX\_NEW/linux/lib/libuclinative.so -Wl,-whole-archive /apps/synopsys/VCSMX\_NEW/linux/lib/libvcsucli.so -Wl,-no-whole-archive /apps/synopsys/VCSMX\_NEW/linux/lib/vcs\_save\_restore\_new.o /apps/synopsys/VCSMX\_NEW/linux/lib/ctype-stubs\_32.a -ldl -lc -lm -lpthread -ldl

../simv up to date

CPU time: .067 seconds to compile + .011 seconds to elab + .072 seconds to link

-----------------------------------------------------------------------------

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; Dec 7 22:12 2014

0 mem[80]=15 mem[81]=35 mem[82]=12

510 mem[80]=5a mem[81]=35 mem[82]=12

1110 mem[80]=5a mem[81]=6a mem[82]=12

1710 mem[80]=5a mem[81]=6a mem[82]=32

2430 mem[80]=9f mem[81]=6a mem[82]=32

3030 mem[80]=9f mem[81]=9f mem[82]=32

3630 mem[80]=9f mem[81]=9f mem[82]=12

$finish called from file "/home/wa/wali9257/ee271/s8sp/implementation/rtl\_vcs/../../s8sp\_tb/tb\_s8sp.v", line 128.

$finish at simulation time 4000000

V C S S i m u l a t i o n R e p o r t

Time: 4000000 ps

CPU Time: 0.140 seconds; Data structure size: 0.0Mb

Sun Dec 7 22:12:54 2014

**---------------------------------------------------------------------------------------------------------------------**

## C.2 Reports (contents) from Netlist (Post-synthesis) Simulations (VCS or NCVERILOG)

Chronologic VCS (TM)

Version I-2014.03-2 -- Sun Dec 7 23:21:23 2014

Copyright (c) 1991-2014 by Synopsys Inc.

ALL RIGHTS RESERVED

This program is proprietary and confidential information of Synopsys Inc.

and may be used and disclosed only as authorized in a license agreement

controlling such use and disclosure.

Parsing design file 's8sp\_netlist.v'

Parsing design file 'tb\_s8sp.v'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CAN2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CAN2XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CAN3X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CAN3XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR11X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR11XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR1X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR1X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR1XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR2X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CANR2XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CAOR2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CDLY1XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CENX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CENX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CENXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CEO3XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CEOX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CEOX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CEOXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CFD1QX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CFD1QX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CFD1QX4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CFD1QXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CHA1X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CHA1XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVDX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVDX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVDXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX16.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX20.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX3.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVX8.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CIVXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CMXI2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CMXI2X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2IX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2IX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2IX4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2IXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2X4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND2XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND3X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND3X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND3XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND4X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CND4X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNIVX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNIVX4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNIVXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2IX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2IX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2IX4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2IXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2X4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR2XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR3X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CNR3X4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND11X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND1X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND1XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND2X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND2XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND3X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND3XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND4CX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COND4CXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COR2X1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COR2X2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/COR2XL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CTSX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CTSX3.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/CTSXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/TMUX21INVprim.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/tsbCFD1QX1.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/tsbCFD1QX2.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/tsbCFD1QX4.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/tsbCFD1QXL.tsbvlibp'

Parsing library directory file '/apps/toshiba/sjsu/verilog/tc240c/TFDPNOprim.tsbvlibp'

Top Level Modules:

tb\_s8sp

TimeScale is 1 ns / 1 ps

Starting vcs inline pass...

49 modules and 2 UDPs read.

However, due to incremental compilation, no re-compilation is necessary.

rm -f \_csrc\*.so linux\_scvhdl\_\*.so pre\_vcsobj\_\*.so share\_vcsobj\_\*.so

ld -m elf\_i386 -shared -o .//../simv.daidir//\_csrc0.so SIM\_l.o 5NrI\_d.o 5NrIB\_d.o

ld -m elf\_i386 -shared -o .//../simv.daidir//pre\_vcsobj\_1\_1.so --whole-archive pre\_vcsobj\_1\_1.a --no-whole-archive

if [ -x ../simv ]; then chmod -x ../simv; fi

g++ -o ../simv -m32 -m32 -Wl,-rpath-link=./ -Wl,-rpath='$ORIGIN'/simv.daidir/ -Wl,-rpath='$ORIGIN'/simv.daidir//scsim.db.dir \_csrc0.so pre\_vcsobj\_1\_1.so rmapats\_mop.o rmapats.o rmar.o /apps/synopsys/VCSMX\_NEW/linux/lib/libzerosoft\_rt\_stubs.so /apps/synopsys/VCSMX\_NEW/linux/lib/libvirsim.so /apps/synopsys/VCSMX\_NEW/linux/lib/librterrorinf.so /apps/synopsys/VCSMX\_NEW/linux/lib/libsnpsmalloc.so /apps/synopsys/VCSMX\_NEW/linux/lib/libvcsnew.so /apps/synopsys/VCSMX\_NEW/linux/lib/libuclinative.so -Wl,-whole-archive /apps/synopsys/VCSMX\_NEW/linux/lib/libvcsucli.so -Wl,-no-whole-archive /apps/synopsys/VCSMX\_NEW/linux/lib/vcs\_save\_restore\_new.o /apps/synopsys/VCSMX\_NEW/linux/lib/ctype-stubs\_32.a -ldl -lm -lc -lpthread -ldl

../simv up to date

CPU time: .191 seconds to compile + .012 seconds to elab + .098 seconds to link + 2.477 seconds in simulation

Chronologic VCS simulator copyright 1991-2014

Contains Synopsys proprietary information.

Compiler version I-2014.03-2; Runtime version I-2014.03-2; Dec 7 23:16 2014

VCD+ Writer I-2014.03-2 Copyright (c) 1991-2014 by Synopsys Inc.

The file '/home/wa/wali9257/ee271/s8sp/implementation/netlist\_vcs/inter.vpd' was opened successfully.

dve> run

0 mem[80]=05 mem[81]=25 mem[82]=54

198 mem[80]=zZ mem[81]=25 mem[82]=54

198 mem[80]=4a mem[81]=25 mem[82]=54

438 mem[80]=4a mem[81]=zZ mem[82]=54

438 mem[80]=4a mem[81]=5a mem[82]=54

678 mem[80]=4a mem[81]=5a mem[82]=zZ

678 mem[80]=4a mem[81]=5a mem[82]=f0

1086 mem[80]=zZ mem[81]=5a mem[82]=f0

1086 mem[80]=8f mem[81]=5a mem[82]=f0

1326 mem[80]=8f mem[81]=zZ mem[82]=f0

1326 mem[80]=8f mem[81]=8f mem[82]=f0

1566 mem[80]=8f mem[81]=8f mem[82]=zZ

1566 mem[80]=8f mem[81]=8f mem[82]=54

1974 mem[80]=zZ mem[81]=8f mem[82]=54

1974 mem[80]=d4 mem[81]=8f mem[82]=54

$finish called from file "tb\_s8sp.v", line 128.

$finish at simulation time 2000000

Simulation complete, time is 2000000 ps.

V C S S i m u l a t i o n R e p o r t

Time: 2000000 ps

CPU Time: 0.040 seconds; Data structure size: 0.2Mb

Sun Dec 7 23:17:17 2014

## C.3 Reports (contents) from Synthesis (Design Compiler)

### C.3.1 Area Report

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report : area

Design : s8sp

Version: C-2009.06-SP5

Date : Sun Dec 7 23:43:24 2014

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Library(s) Used:

tc240c (File: /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_WCCOM25)

Number of ports: 20

Number of nets: 49

Number of cells: 2

Number of references: 2

Combinational area: 863.500000

Noncombinational area: 464.500000

Net Interconnect area: undefined (No wire load specified)

Total cell area: 1328.000000

Total area: undefined

1

### C.3.2 Power Report

Loading db file '/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_WCCOM25'

Loading db file '/apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_BCCOM25'

Warning: Main library 'tc240c' does not specify the following unit required for power: 'Leakage Power'. (PWR-424)

Information: Propagating switching activity (low effort zero delay simulation). (PWR-6)

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report : power

-analysis\_effort low

Design : s8sp

Version: C-2009.06-SP5

Date : Sun Dec 7 23:43:25 2014

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Library(s) Used:

tc240c (File: /apps/toshiba/sjsu/synopsys/tc240c/tc240c.db\_WCCOM25)

Operating Conditions: WCCOM25 Library: tc240c

Wire Load Model Mode: top

Global Operating Voltage = 2.3

Power-specific unit information :

Voltage Units = 1V

Capacitance Units = 1.000000ff

Time Units = 1ns

Dynamic Power Units = 1uW (derived from V,C,T units)

Leakage Power Units = Unitless

Cell Internal Power = 1.8601 mW (95%)

Net Switching Power = 100.8653 uW (5%)

---------

Total Dynamic Power = 1.9610 mW (100%)

Cell Leakage Power = 0.0000

1

-----------------------------------------------------------------------------

### C.3.3 Timing Report

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report : timing

-path full

-delay max

-group clk

-max\_paths 10

Design : s8sp

Version: C-2009.06-SP5

Date : Sun Dec 7 23:43:24 2014

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Operating Conditions: WCCOM25 Library: tc240c

Wire Load Model Mode: top

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_ir\_instance/ir\_out\_reg[4]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 f

reset (in) 0.00 0.00 f

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 f

u1\_data\_block\_instantiation/u1\_ir\_instance/reset (local\_ir\_reg)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_ir\_instance/U2/Z (CIVX16)

0.04 0.04 r

u1\_data\_block\_instantiation/u1\_ir\_instance/U3/Z (CND2X2)

0.11 0.15 f

u1\_data\_block\_instantiation/u1\_ir\_instance/U5/Z (CND2X1)

0.07 0.22 r

u1\_data\_block\_instantiation/u1\_ir\_instance/U4/Z (COND1XL)

0.17 0.38 f

u1\_data\_block\_instantiation/u1\_ir\_instance/ir\_out\_reg[4]/D (CFD1QX4)

0.00 0.38 f

data arrival time 0.38

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.36 0.39

data required time 0.39

--------------------------------------------------------------------------

data required time 0.39

data arrival time -0.38

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_gr\_instance/gr\_on\_data\_reg[6]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 r

reset (in) 0.00 0.00 r

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 r

u1\_data\_block\_instantiation/u1\_gr\_instance/reset (gen\_reg)

0.00 0.00 r

u1\_data\_block\_instantiation/u1\_gr\_instance/U11/Z (CIVX3)

0.04 0.04 f

u1\_data\_block\_instantiation/u1\_gr\_instance/U26/Z (CND3XL)

0.14 0.18 r

u1\_data\_block\_instantiation/u1\_gr\_instance/U32/Z (COND3X1)

0.21 0.39 f

u1\_data\_block\_instantiation/u1\_gr\_instance/gr\_on\_data\_reg[6]/D (CFD1QX2)

0.00 0.39 f

data arrival time 0.39

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.36 0.39

data required time 0.39

--------------------------------------------------------------------------

data required time 0.39

data arrival time -0.39

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: dat[3] (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_gr\_instance/gr\_on\_data\_reg[3]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 f

dat[3] (inout) 0.00 0.00 f

u1\_data\_block\_instantiation/data\_bus[3] (data\_block)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_gr\_instance/data\_on\_gr[3] (gen\_reg)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_gr\_instance/U13/Z1 (CIVDX1)

0.12 0.12 f

u1\_data\_block\_instantiation/u1\_gr\_instance/U20/Z (CAOR2X1)

0.28 0.40 f

u1\_data\_block\_instantiation/u1\_gr\_instance/gr\_on\_data\_reg[3]/D (CFD1QX2)

0.00 0.40 f

data arrival time 0.40

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.35 0.40

data required time 0.40

--------------------------------------------------------------------------

data required time 0.40

data arrival time -0.40

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_gr\_instance/gr\_on\_data\_reg[4]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 f

reset (in) 0.00 0.00 f

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 f

u1\_data\_block\_instantiation/u1\_gr\_instance/reset (gen\_reg)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_gr\_instance/U18/Z (COR2X1)

0.24 0.24 f

u1\_data\_block\_instantiation/u1\_gr\_instance/U30/Z (COND3X1)

0.17 0.42 r

u1\_data\_block\_instantiation/u1\_gr\_instance/gr\_on\_data\_reg[4]/D (CFD1QX1)

0.00 0.42 r

data arrival time 0.42

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.33 0.42

data required time 0.42

--------------------------------------------------------------------------

data required time 0.42

data arrival time -0.42

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/flag\_reg\_reg[1]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 f

reset (in) 0.00 0.00 f

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 f

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/reset (flag\_reg)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U5/Z (CIVX12)

0.04 0.04 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U12/Z (CND2X2)

0.10 0.14 f

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U15/Z (CNR2IX1)

0.12 0.25 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U14/Z (CND2IX1)

0.18 0.43 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/flag\_reg\_reg[1]/D (CFD1QX1)

0.00 0.43 r

data arrival time 0.43

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.32 0.43

data required time 0.43

--------------------------------------------------------------------------

data required time 0.43

data arrival time -0.43

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_ir\_instance/ir\_out\_reg[6]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 f

reset (in) 0.00 0.00 f

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 f

u1\_data\_block\_instantiation/u1\_ir\_instance/reset (local\_ir\_reg)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_ir\_instance/U2/Z (CIVX16)

0.04 0.04 r

u1\_data\_block\_instantiation/u1\_ir\_instance/U6/Z (CND2X2)

0.11 0.15 f

u1\_data\_block\_instantiation/u1\_ir\_instance/U12/Z (CND2XL)

0.09 0.25 r

u1\_data\_block\_instantiation/u1\_ir\_instance/U11/Z (CND2XL)

0.14 0.39 f

u1\_data\_block\_instantiation/u1\_ir\_instance/ir\_out\_reg[6]/D (CFD1QX4)

0.00 0.39 f

data arrival time 0.39

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.36 0.39

data required time 0.39

--------------------------------------------------------------------------

data required time 0.39

data arrival time -0.39

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/flag\_reg\_reg[3]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 r

reset (in) 0.00 0.00 r

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/reset (flag\_reg)

0.00 0.00 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U2/Z (CIVX20)

0.03 0.03 f

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U9/Z (CND2IX4)

0.06 0.09 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U10/Z (CND2X2)

0.08 0.16 f

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/U13/Z (COND11X1)

0.24 0.40 r

u1\_data\_block\_instantiation/u1\_flag\_reg\_instance/flag\_reg\_reg[3]/D (CFD1QX2)

0.00 0.40 r

data arrival time 0.40

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.35 0.40

data required time 0.40

--------------------------------------------------------------------------

data required time 0.40

data arrival time -0.40

--------------------------------------------------------------------------

slack (MET) 0.00

Startpoint: dat[5] (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_ar\_instance/ar\_on\_bus\_reg[5]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 r

dat[5] (inout) 0.00 0.00 r

u1\_data\_block\_instantiation/data\_bus[5] (data\_block)

0.00 0.00 r

u1\_data\_block\_instantiation/U9/Z (CIVX8) 0.03 0.03 f

u1\_data\_block\_instantiation/U71/Z (CIVX4) 0.07 0.10 r

u1\_data\_block\_instantiation/u1\_ar\_instance/data\_on\_ar[5] (addr\_reg)

0.00 0.10 r

u1\_data\_block\_instantiation/u1\_ar\_instance/U10/Z (CIVXL)

0.10 0.20 f

u1\_data\_block\_instantiation/u1\_ar\_instance/U9/Z (COND2X1)

0.20 0.40 r

u1\_data\_block\_instantiation/u1\_ar\_instance/ar\_on\_bus\_reg[5]/D (CFD1QX2)

0.00 0.40 r

data arrival time 0.40

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.34 0.41

data required time 0.41

--------------------------------------------------------------------------

data required time 0.41

data arrival time -0.40

--------------------------------------------------------------------------

slack (MET) 0.01

Startpoint: dat[5] (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_dr\_instance/dr\_on\_data\_reg[5]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 r

dat[5] (inout) 0.00 0.00 r

u1\_data\_block\_instantiation/data\_bus[5] (data\_block)

0.00 0.00 r

u1\_data\_block\_instantiation/U9/Z (CIVX8) 0.03 0.03 f

u1\_data\_block\_instantiation/U71/Z (CIVX4) 0.07 0.10 r

u1\_data\_block\_instantiation/u1\_dr\_instance/data\_on\_dr[5] (data\_reg)

0.00 0.10 r

u1\_data\_block\_instantiation/u1\_dr\_instance/U16/Z (CIVXL)

0.10 0.20 f

u1\_data\_block\_instantiation/u1\_dr\_instance/U15/Z (COND2X1)

0.20 0.40 r

u1\_data\_block\_instantiation/u1\_dr\_instance/dr\_on\_data\_reg[5]/D (CFD1QX2)

0.00 0.40 r

data arrival time 0.40

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.34 0.41

data required time 0.41

--------------------------------------------------------------------------

data required time 0.41

data arrival time -0.40

--------------------------------------------------------------------------

slack (MET) 0.01

Startpoint: reset (input port)

Endpoint: u1\_data\_block\_instantiation/u1\_ir\_instance/ir\_out\_reg[5]

(rising edge-triggered flip-flop clocked by clk)

Path Group: clk

Path Type: max

Point Incr Path

--------------------------------------------------------------------------

input external delay 0.00 0.00 f

reset (in) 0.00 0.00 f

u1\_data\_block\_instantiation/reset (data\_block) 0.00 0.00 f

u1\_data\_block\_instantiation/u1\_ir\_instance/reset (local\_ir\_reg)

0.00 0.00 f

u1\_data\_block\_instantiation/u1\_ir\_instance/U2/Z (CIVX16)

0.04 0.04 r

u1\_data\_block\_instantiation/u1\_ir\_instance/U6/Z (CND2X2)

0.11 0.15 f

u1\_data\_block\_instantiation/u1\_ir\_instance/U22/Z (CND2X1)

0.08 0.24 r

u1\_data\_block\_instantiation/u1\_ir\_instance/U7/Z (COND1X1)

0.14 0.38 f

u1\_data\_block\_instantiation/u1\_ir\_instance/ir\_out\_reg[5]/D (CFD1QX4)

0.00 0.38 f

data arrival time 0.38

max\_delay 1.00 1.00

clock uncertainty -0.25 0.75

library setup time -0.36 0.39

data required time 0.39

--------------------------------------------------------------------------

data required time 0.39

data arrival time -0.38

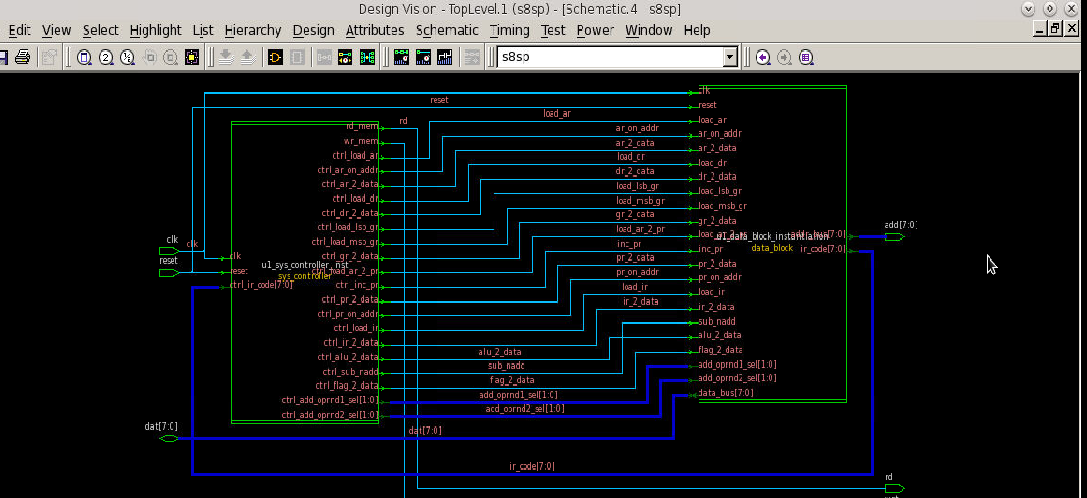
--------------------------------------------------------------------------

slack (MET) 0.01

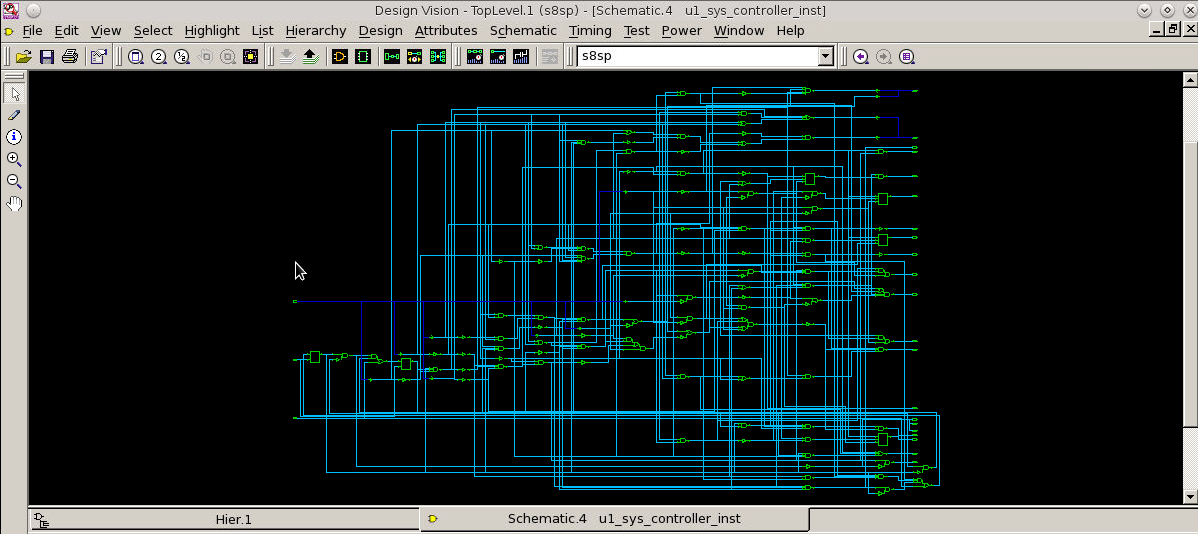
1

## C.4 Screenshot Circuits from Synthesis (Design Compiler)

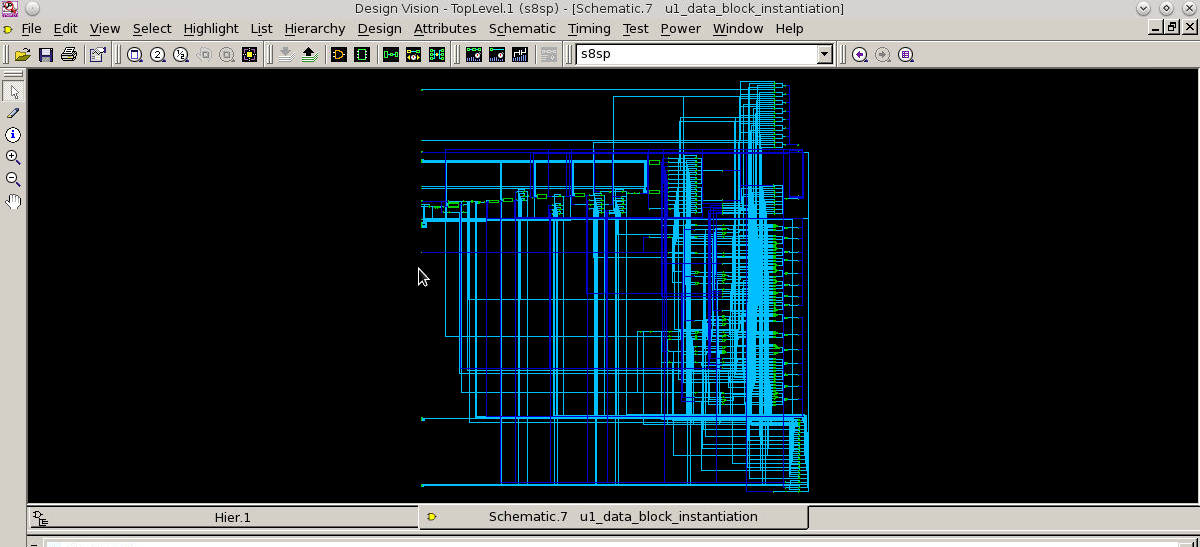
### C.4.1 TopLevel Schematic of the design



### C.4.2 Controller Schematic



### C.4.3 Data\_Block Schematic

****