

CSE 140L Hardware Lab Four, Winter 2006

System Design using Datapath and Control Subsystems

Thomas Y. Lee

Due time: March 15, 2006

Objective

In this lab, we will build a simple system using datapath, memory, registers, control logic and a program counter. We use buses and standard modules as basic building blocks for datapath, memory, registers and program counter design, and use VHDL code for control subsystem.

Note that in this lab, we use Altera FPGA Family: Stratix, Device Type: EP1S, Package Type: 144 pin TQFP

The Project

Design a simple system that can perform addition, shifting, comparison, AND, OR, XOR and NOT. The system reads an 8-bits value from a read-only memory (ROM) block (4-bit operation code & 4-bit data). The memory block should be able to hold at least 16 instructions. There are three 4-bit registers R1, R2, and R3. Register R1 and R2 store the operands. Register R3 presents the result of the calculations. There are also a 1-bit flag for overflow and another 1-bit flag for comparison. We list the specification of the instructions:

Command	4-bit Inst.	4-bit Data	Description
1 Init	0000	XXXX	Initialize all registers, ie $R1=R2=R3=(0,0,0,0)$
2 Move1	0001	aaaa	Put aaaaa value into register R1
3 Move2	0010	bbbb	Put bbbbb value into register R2
4 Store	0011	XXXX	Take the previous result in R3 and store into register R2
5 Add	0100	XXXX	Add contents in R1 and R2. Present the result at R3 and output overflow flag

6 Shift Left(SHL)	0101	aaXX	Left shift the content in R2 by aa bits (no rotate), then store the result in R3
7. Shift Right (SHR)	0110	aaXX	Right shift the content in R2 by aa bits (no rotate) then store the result in R3
8 Comp	0111	XXXX	Compare the contents of R1 and R2. The flag bit is false if $R2 > R1$. or $R2 < R1$, This flag is true if $R2 = R1$
9 MASK(AND)	1000	XXXX	Mask (AND-ing) R1 and R2. Present the result in R3
10 OR	1010	XXXX	Perform OR function on the contents in R1 and R2. Present the result at R3
11 XOR	1011	XXXX	Perform XOR function on the contents in R1 and R2. Present the result at R3
12.NOT	1100	XXXX	Perform NOT function on the contents in R1 Present the result at R3
13. NOP	1101	XXXX	Perform No Operation

*For 4-bit data, X is Don't Care.

**Note that register R1 holds its content except for instructions 1 and 2. Register R2 holds except for instructions 1, 3, and 4. Register R3 holds except for instructions 1, 5, 6, 7, 9, 10, 11 and 12. In addition, a program counter and control subsystem are needed to make the whole system works.

Report

Title page:

- Names of students and due date.
- Title of the lab and objective.
- A brief description of each person's contribution.

Contents:

1. Schematic diagrams of the top-level design, memory, datapath, registers and program counter - must use bus.
2. VHDL or Verilog code of the control subsystem using one truth table.
3. Layout of the design (FPGA)
4. Your instructions in the memory
5. Functional simulation of your instructions. Demonstrate the content of the output of each of the 16 instructions. Label and explain the data before and after each operation.
6. Discussion of the verification. Show the reason why we are confident that the system can function properly. List the process needed for the proof.

7. Illustrate the worst delay path between registers with timing simulation.
8. Execute the following simple Assembly Language in the processor that you Design:

```
Init
Move1 0010
Move2 0011
NOT
STORE
XOR
SHR 01XX
Move1 1010
ADD
Comp
NOP
```

Show the final results of R1, R2, R3, overflow flag and comparison flag

Grading

60% will be based on the completeness and correctness of the report.
40% will be based on the demonstration with TA.