SPIM16 EECS Project 2 Specification

REGISTER FILE INFORMATION

Register (16 bit)

- Circuit: Reg16
- Inputs:
  - ld
  - clr
  - clk
  - D[15:0]
- Outputs:
  - Q[15:0]

NOTE: Use D-Flip-Flops and not Registers.

Register File

- Circuit: RegFile
- Inputs:
  - clr
  - clk
  - D[15:0]
  - Asel[3:0]
  - Bsel[3:0]
  - Dsel[3:0]
- Outputs:
  - A[15:0]
  - B[15:0]
- Sub-Circuit: Reg16 x16

NOTES: 16x16-bit register file, 2 read ports and 1 write port. Register 0 always reads as 0. Use Reg16s and not Registers.

ARITHMETIC/LOGIC UNIT

ALU

- Circuit: ALU
- Inputs:
  - Cin
  - A[15:0]
  - B[15:0]
  - ALUctrl[2:0]
- Outputs:
  - R[15:0]
  - Cout
- Sub-Circuit: Shifter x2

ALUctrl	Inst.	Operation
0 0 0	`or`	R = A | B
0 0 1	`xor`	R = A ^ B
0 1 0	`and`	R = A & B
0 1 1	`add`	Cout, R = A + B + Cin
1 0 0	`sub`	Cout, R = A – B
1 0 1	`sll`	R = A << B
1 1 0	`srl`	R = A >> B
1 1 1	`sra`	R = A >>> B

Shifter x2 (`sll, sr(l/a)`)

- Circuit: ShiftLeft, ShirtRight
- Inputs:
  - D[15:0]
  - ShiftAmt[3:0]
  - SignExt (ShiftRight)
- Outputs:
  - R[15:0]

CONTROL UNIT

Main Decoder

- Circuit: MainDec
- Inputs:
  - OpCode[3:0]
- Outputs:
  - MemRead
  - MemWrite
  - ALUop
  - Function[2:0]
  - ???

ALUop	Meaning
0	Add
1	Use function

ALU Decoder

- Circuit: AluDec
- Inputs:
  - ALUop
  - Function[2:0]
- Outputs:
  - ALUctrl[2:0]
  - ???

Control Unit

- Circuit: ControlUnit
- Inputs: ???
- Outputs: ???
- Sub-Circuit: MainDec
- Sub-Circuit: AluDec

HAZARD CONTROL UNIT

Hazard Control

- Circuit: HazardCtrl
- Inputs: ???
- Outputs: ???

SPIM16 EECS Project 2 Specification

CPU INFORMATION

Central Processing Unit

- Circuit: CPU
- Inputs:
  - power
  - clk
  - Input[15:0]
- Outputs:
  - Output[15:0]
  - Address[15:0]
- Sub-Circuit: RegFile
- Sub-Circuit: ALU
- Sub-Circuit: ControlUnit
- Sub-Circuit: HazardCtrl
- ROM: Instruction Memory
- RAM: Data Memory

NOTE: Use a standard 5-stage pipeline.

MEMORY ALLOCATION

Addresses	Allocation
0x7FFF – 0x1000	Stack Data
0x1000 – 0x7FFF	Heap Data
0x0000 – 0x0FFF	Text/Static Data

NOTE: Memory is byte addresses and word aligned with a data and address width of 16 bits. Use synchronous load/store ports with a separate ports for load and store.

DATA ALIGNMENT

WORD			
BYTE 0		BYTE 1	
NIBBLE	NIBBLE	NIBBLE	NIBBLE

NOTE: Be careful of the data alignment when moving data from or to system memory.

*** OPTIONAL SECTION ***

I/O INFORMATION

Memory Mapped I/O

- Circuit: MMIO
- Inputs:
  - Address[15:0]
  - CPUin[15:0]
  - MEMin[15:0]
  - KEYin[15:0]
  - read
  - write
- Outputs:
  - CPUout[15:0]
  - MEMout[15:0]
  - TERMout[15:0]

MMIO

Addresses	Device	Circuit
0x0000 – 0x7FFF	Data Memory	MEM
0xA000 – 0xAFFF	Keyboard	KEY
0xB000 – 0xBFFF	Terminal Window	TERM

Keyboard

Local Addresses	Port/Pins	I/O
0x0000	Data[7:0]	Output
0x0002	Available	Output
0x0004	Clear	Input

Terminal Window

Local Addresses	Port/Pins	I/O
0x0000	Data[7:0]	Input
0x0002	Clear	Output

Keyboard

- Circuit: KEY
- Inputs:
  - KEYaddr[15:0]
  - clk
  - read
- Outputs:
  - KEYdata[15:0]

Terminal Window

- Circuit: TERM
- Inputs:
  - TERMaddr[15:0]
  - TERMdata[15:0]
  - clk
  - write