HW#8 - key

4.5 Single-cycle datapath. Assume that the machine instruction fetched is 0x00c6ba23
Refer to Figure 4.21 for this problem

given machine instruction: (see page 125 for function codes)

000000 00110 00110 10111 01000 100011 Assembly: subu rd, rs, rt

opcode rs rt rd shift function Meaning: rd = rs - rt

(Note: shifting is not done by the subu instruction.)

4.5.1 ALU control unit inputs:

1) ALU Op - two bits, will be 10

2) Function field of the instructions, 6 bits, will be 100011 for the subu instruction (ALU control output =0110)

4.5.2 New PC address, i.e. PC value for the next instruction following subu

PC+4 PC reg output goes to one input on the adder+4, then routes to input 0 on a mux and hence to PC reg input

4.5.3 For each mux, detail the values of its input and outputs while executing this instruction.

Register file write address mux

port 0: bits 20-16 of the instruction

port 1: bits 15-11 of the instruction

output: bits 15-11 of the instruction

control: RegDst = 1

ALU source mux

port 0: Data from Register File output 2, which is R[6] contents

port 1: Sign extended lower 16 bits of the instruction

output: Data from Register File output 2, which is R[6]; goes to ALU in B

control: ALUsrc = 0

Write-back mux (MemToReg)

port 0: ALU output

port 1: Data word read from memory

output: ALU output

control: MemtoReg = 0

The PC source mux

port 0: has PC+4 as the input

port 1: address generator output

output: PC+4

control: Branch = 0

4.5.4 What are the input values for the ALU and the two adder units?

ALU input A: has R[6] routed to it directly

ALU input B: has R[6] routed to it via the mux

The PC+4 adder input A: current PC value

The PC+4 adder input B: a value of +4

Branching adder input A: PC+4

Branching adder input B: sign extended bits 0-15 shifted left two

4.5.5 What are the values of all inputs for the register file?

Read register-1: 6 RegWrite: 1
Read register-2: 6
Write register : 23
Write data : ALU result arriving via the Write-back Mux

4.6 Operation of I-type instructions such as addi or andi

4.6.1 What logic blocks, if any, are needed to add I-type instructions to figure 4.21? Add any needed logic blocks and explain their purpose.

No additional block or control signals needed for addi or addiu
For andi or ori however, a new control signal ZeroExtend is needed to select extending zeros rather than the sign bit of the immediate value.

4.6.2 List the values of the signals generated by the control unit for addi. Explain any "don't care" control signals.

RegDest: 0
Branch: 0
MemRead: 0 (could be don't care since we are not using data from this memory)
MemToReg: 0
ALUop: 10 (two bits)
MemWrite: 0
ALUsrc: 1
RegWrite: 1

4.7 Refer to figure 4.21

memory	250ps	adder	150ps	sign extend	50ps
register file	150ps	a gate	5ps	control	50ps
mux	25ps	register read	30ps		
alu	200ps	register setup	20ps		

4.7.1 What is the latency of an R-type instruction?

PC + Mem+RegFile+Mux+ALU+Mux+RegSetup
30 + 250 + 150 + 25 + 200 + 25 + 20 = 700ps

4.7.2 What is the latency of lw?

PC + Mem+RegFile+Mux+ALU+ Mem+mux+RegSetup
30 + 250 + 150 + 25 + 200 + 250 + 25 + 20 = 950ps