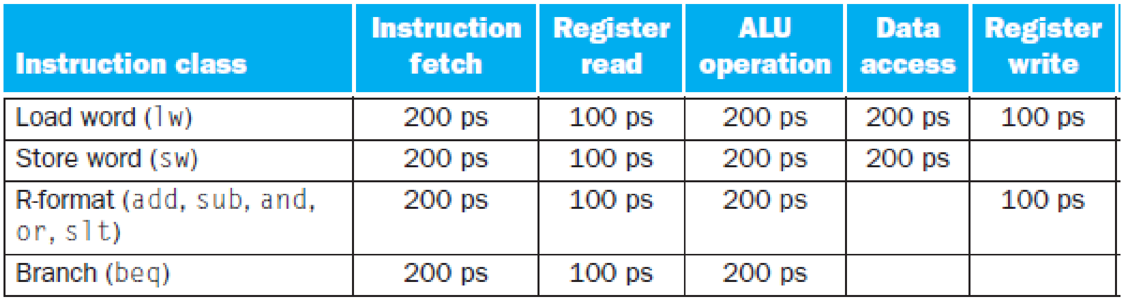
EEL4768 Computer Architecture

Homework 4

Due date: April 9, 2017, 11:59PM (late submission is not accepted)

Total points: 100

1. (20 points) Answer the following questions based on the following table. All different datapaths support the following four instruction types with listed delay of each component.



Instruction class	Instruction fetch	Register read	ALU operation	Data access	Register write
Load word (lw)	200 ps	100 ps	200 ps	200 ps	100 ps
Store word (sw)	200 ps	100 ps	200 ps	200 ps	
R-format (add, sub, and, or, slt)	200 ps	100 ps	200 ps		100 ps
Branch (beq)	200 ps	100 ps	200 ps		

a) What should clock cycle time be for single-cycle datapath, multi-cycle datapath and pipelined datapath? Please explain your answer.

b) What is the latency (defined as the delay from when the instruction enters the datapath until it finishes) of R-format instruction in single-cycle datapath, multi-cycle datapath and pipelined datapath? Assume no hazard is detected in pipelined datapath. Please explain your answer.

c) What is the latency (defined as the delay from when the instruction enters the datapath until it finishes) of load word (lw) instruction in single-cycle datapath, multi-cycle datapath and pipelined datapath? Assume no hazard is detected in pipelined datapath. Please explain your answer.

d) What is the averaged throughput (defined as the number of instructions executed in **1 nano-second in this case**) of single-cycle datapath, multi-cycle datapath and pipelined datapath? Assume the frequency of four different instruction types are the same. Assume no hazard is detected in pipelined datapath. Please explain your answer.

2. (10 points) Suppose that an unpipelined processor has a cycle time of 25 ns, and its datapath consists of modules with the latencies of 2, 3, 4, 7, 3, 2, and 4 ns (in this specific order). In pipelining this processor, it is not possible to rearrange the order of the modules (for example, putting the register read stage before the instruction decode stage) or to divide a module into multiple pipeline stages (for complexity reasons). Note that, this is not standard five-stage pipeline design.

a) What is the minimum clock cycle time that can be achieved by pipelining this processor? Please explain your answer

b) If you are limited to a 2-stage pipeline, what is the minimum clock cycle time? Please explain your answer.

3. (20 points) The following codes run on the five-stage pipelined datapath. For each piece of code, answer the following questions.

a) Without forwarding, insert the necessary number of 'nops' for the code to execute correctly

b) With the forwarding unit available (only supports forwards MEM-> EX and WB->EX), show how the code will execute. Mention the data that's forwarded between the stages. **Use 'nops' only when necessary.**

Code 1:

```
sub t1, t2, t3

add t1, t4, t5

or t4, t2, t6
```

Code 2:

```
and t2, t5, t1

sub t3, t2, t0

nor t7, t1, t5
```

Code 3:

```
and t2, t5, t1

sub t3, t2, t0

nor t7, t2, t1
```

Code 4:

```
lw t1, 22(t0)
and t2, t1, t3
```

Code 5:

```
lw t1, 22(t0)
sub t6, t0, t2
xor t3, t1, t5
```

4. (10 points) For standard five-stage pipeline datapath

a) Explain the stall-on-branch strategy for implementing the 'beq' instruction. Explain this strategy for the two approaches of implementing the 'beq' (in ID stage and in MEM stage). Answer the items in the list below.

b) Explain the branch prediction strategy for implementing the ‘beq’ instruction. Explain this strategy for the two approaches of implementing the ‘beq’ (in ID stage and in MEM stage). Answer the items in the list below.

c) Explain the delayed branch strategy for implementing the ‘beq’ instruction. Explain this strategy for the two approaches of implementing the ‘beq’ (in ID stage and in MEM stage). Answer the items in the list below.

- How many ‘nops’ are used if the branch is taken?
- How many ‘nops’ are used of the branch is not taken?
- Is this solution always applicable? Explain.

5. (10 points) For five-stage pipelined datapath, please **explain each line** of the following code and **fill** “_______”

```
If( EX/MEM.RegWrite==1 and
EX/MEM.RegisterRd != 0 and
EX/MEM.RegisterRd == ID/EX.RegisterRs )

Then, forward from ___________(which stage?) to __________(which
operand of which stage?)
```

6. (10 points) For five-stage pipelined datapath, please explain **each line** of the following code and **fill** “_______”

```
If( MEM/WB.RegWrite==1 and
MEM/WB.RegisterRd != 0 and
EX/MEM.RegisterRd != ID/EX.RegisterRs and
MEM/WB.RegisterRd == ID/EX.RegisterRs )

Then, forward from ___________(which stage?) to __________(which
operand of which stage?)
```

7. (10 points) The figure below shows how the forwarding unit sets the multiplexers to forward from the MEM or WB stages to the EX stage.

a) What are the values of “ForwardA” and “ForwardB” when the “sub” instruction in EX stage:

add $2, $1, $4

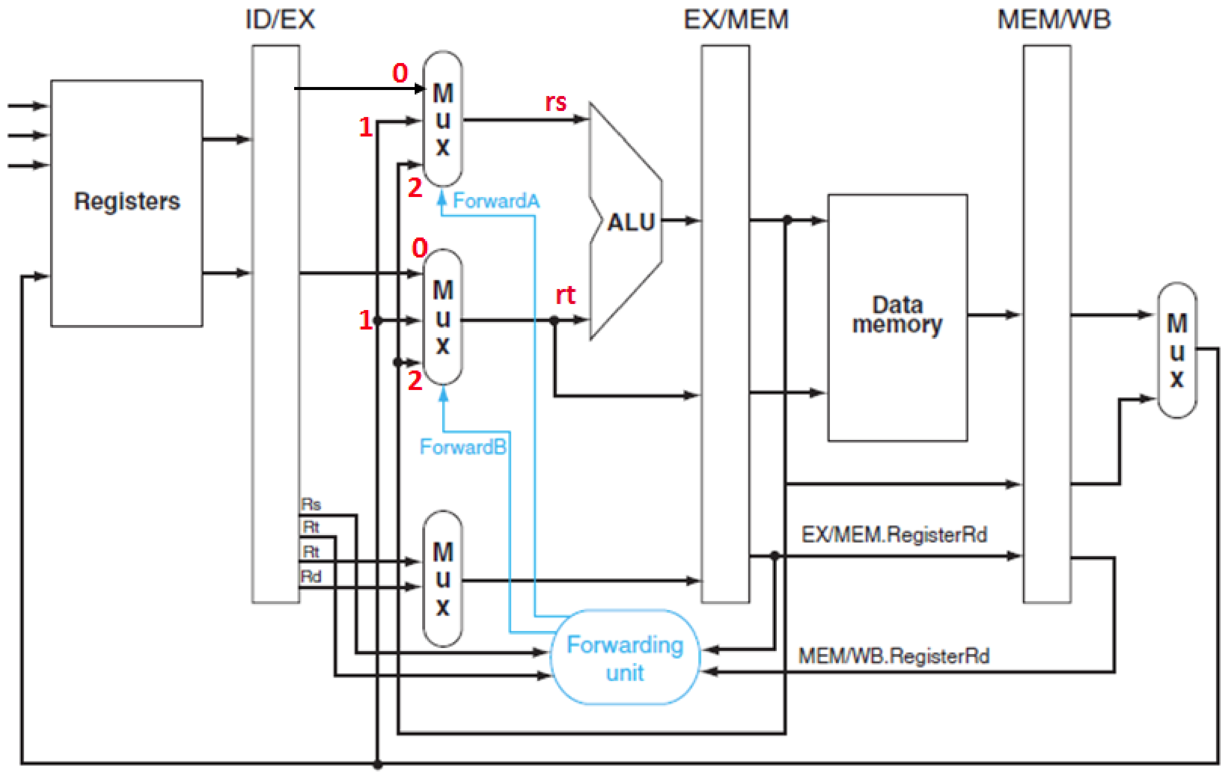
sub $3, $2, $1

b) What are the values of “ForwardA” and “ForwardB” when the “and” instruction in EX stage:

lw $2, 5($0)

sub $1, $3, $4

and $5, $2, $4



8. (10 points) If the ‘beq’ instruction is implemented in the ID stage, how do the codes below execute correctly? We would like to use forwarding to reduce the use of ‘nops’.

a):

sub t1, t2, t3

beq t1, t2, label

b):

sub t1, t2, t3

add t0, t2, t3

beq t1, t2, label

c):

lw t1, 5(s0)

beq t1, t0, label