Cpr E 381 Homework 9

P1. Suppose we want to perform 64 bit arithmetic in MIPS. A 64 bit number is stored in pairs of two registers $reg and $(reg+1) where the least significant 32 bits are in register $(reg+1) and the most significant 32 bits are stored in register $reg. The numbers are signed numbers.

(a) Find a set of two instruction sequence of MIPS instructions to determine if there is a carry out from the addition of two registers $t3 and $t5. Place a zero or one in register $t0 if the carry out is zero or 1, respectively.

(b) Find a set of four instruction sequence of MIPS instructions to perform double precision (64 bits) integer addition. Assume that the two 64bits numbers are stored in registers $t2, $t3 and registers $t4, $t5, respectively. The sum is to be stored in registers $t0, $t1.

(c) Give an example data when your code will fail to produce correct output even when there is no overflow.

P2. The goal of this problem to study performance analysis.

(a) A computer center receives tasks to process in batches. In one batch they received 30 tasks and they processed them at a rate of 60 tasks per hour. There stated goal is that they would like to process 90 tasks per hour. At what rate they should process the next 30 jobs to meet their goal.

(b) Amdahl’s law indicates how the modifications should be carried out to benefit most. Suppose a task takes time T. Suppose a part of the program that can be speeded up is executed in time T’. A new algorithm will increase the performance of this part by a factor of p. Drive an expression for speed up. If you want to achieve a speed up of 5 and T’ = 0.8T, what value of p would achieve the goal?

(c) Bob needs to use a 3-bit down-counter. However, he only has a 4-bit synchronous up-counter with outputs q3-q0 and their complements available. He does not know how to modify the internal structure of the up-counter. How can he construct the 3-bit down-counter using only the device he has?

P3. Consider the datapath for a single cycle implementation shown in the book to implement a set of selected instructions of MIPS 3000 without Jump and JAL.

(a) Show datapath changes necessary to implement the jal (jump-and-link) instruction.

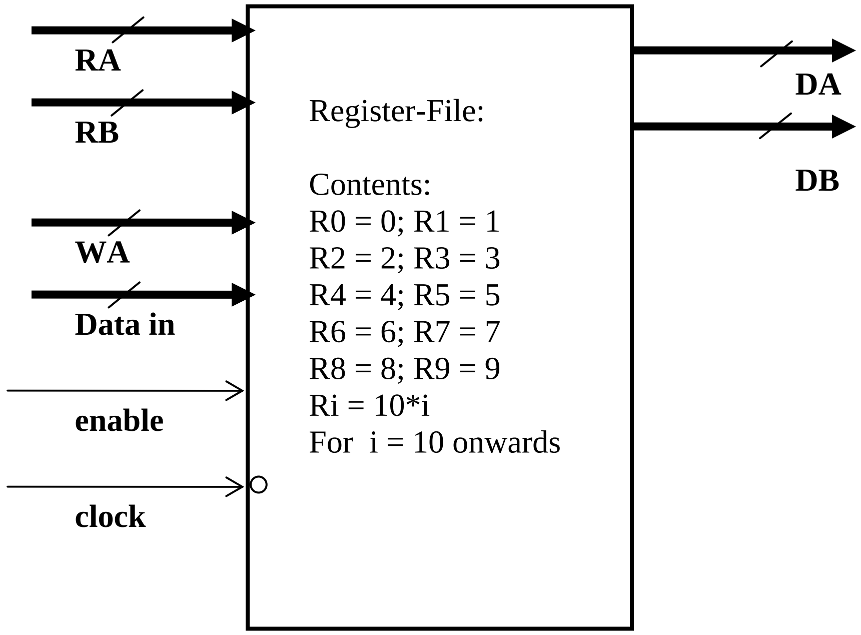
(b) Show all the control signals in a tabular form for all classes of instructions including lw, sw, R-Type, bre, jump and jal instruction..

P4. A 16x16 register file (32 registers with 16 bits each) has been implemented using rising edge triggered D flip-flops with enable signal. If enable signal for a register is 0, it retains its data. Otherwise, it takes new data at the rising edge of the clock. Also,

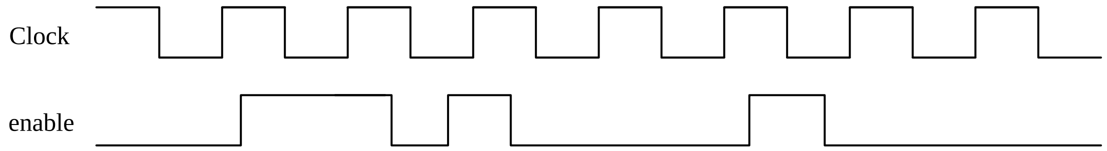
1. **RA** is the read address for port A and output is delivered to data port **DA**.
2. **Rb** is the read address for port B and output is delivered to data port.
3. **WA** is the address of register in which the data at Data in port are written into if **enable** signal is on.

A. Specify the number of bits for each of the inputs RA, RB, WA, and Datain and outputs DA and DB.

B. Draw a picture of rising edge clocking D-flip flop with enable signal.
C. Mark on the figure when data will change, and what register address and data value will be used.
D. Show the data values at the outputs DA and DB for each half clock cycle in the figure below. Assume that the propagation delay is zero.



A16x16 Register file with initial contents



WA	2	2	3	3	8	8	11	11	14	14	6	6	7	7	4	4
Datain	514	57	38	33	88	78	23	12	21	44	46	65	54	73	54	55
RA	2	2	2	2	2	2	2	2	3	3	3	3	3	3	4	4
RB	8	8	8	8	8	8	8	8	6	6	6	6	6	6	4	4
DA																
DB																

P5. Problem 6.1 If the time for an ALU operation an be shortened by 25% (compared to the description in Figure 6.2 on page 373);

a. Will it affect the speedup obtained from pipelining? If yes, by how much? Otherwise, why?
b. What if the ALU operation now takes 25% more time?

P6. Problem 6.2 A computer architect needs to design the pipeline of a new microprocessor. She has an example workload program core with 10^6 instructions. Each instruction takes 100 ps to finish.

a. How long does it take to execute this program core on a non-pipelined processor?
b. The current state of the art microprocessor has about 20 pipeline stages. Assume it is perfectly pipelined. How much speedup will it achieve compared to the non-pipelined processor?
c. Real pipelining isn't perfect, since implementing pipelining introduces some overhead per pipeline stage. Will this overhead affect instruction latency, instruction throughput or both? Explain.

P7. Problem 6.4 Identify all of the data dependencies in the following code. Which dependencies are data hazards that will be resolved via forwarding? Which dependencies are data hazards that will cause a stall?

```
add $3, $4, $2
sub $5, $3, $1
lw  $6, 200($3)
add $7, $3, $6
```

P8. Problem 6.14 The following piece of code is executed using the pipeline shown in Figure 6.30 on page 409:

```
lw  $5, 40($2)
add $6, $3, $2
or  $7, $2, $1
and $8, $4, $3
sub $9, $2, $1
```

At cycle 5, right before the instructions are executed, the processor state is as follows:

a. The PC has the value 100_{ten}, the address of the sub instruction.
b. Every register has the initial value 10_{ten} plus the register number (e.g. register $8 has the value 18_{ten}).
c. Ever memory word accessed as data has the initial value 1000_{ten} plus the byte address of the word (e.g. Memory[8] has the initial value 1008_{ten}).

Determine the value of every field in the four pipeline registers in cycle 5.

P9. Problem 6.21 We have a program of 10^3 instructions in the formant of "lw, add, lw, add, ..." The add instruction depends (and only depends) on the lw instruction right before it. The lw instruction depends (and only depends) on the add instruction right before it. If the program is executed on the pipelined datapath of Figure 6.36 on page 416:

a. What would be the actual CPI?
b. Without forwarding, what would be the actual CPI?

P10. Problem 6.22 Consider executing the following code on the pipelined datapath of Figure 6.36 on page 416.

```
     lw  $4, 100($2)
     sub $6, $4, $3
     add $2, $3, $5
```

How many cycles will it take to execute this code? Draw a diagram like that of Figure 6.34 on page 414 that illustrates the dependencies that need to be resolved, and provide another diagram like that of Figure 6.35 on page 415 that illustrates how the code will actually be executed (incorporating any stalls or forwarding) so as to resolve the identified problems.