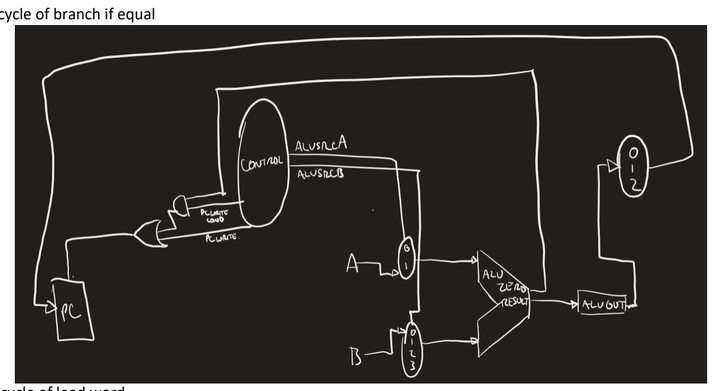
Lab 7 – Practice for CPU implementation

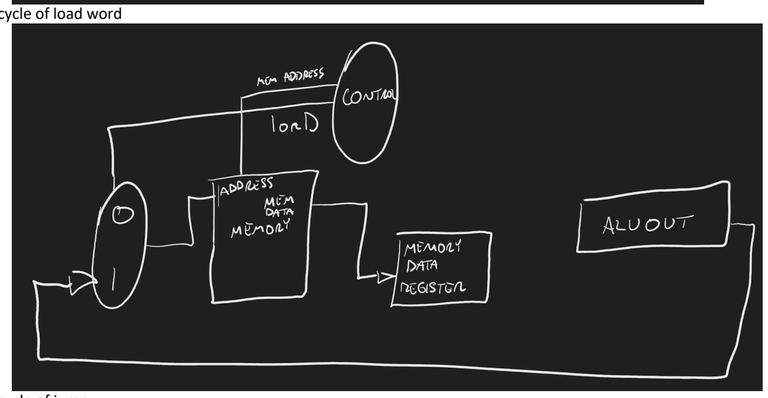
Tuesday, April 6, 2021 12:20 PM

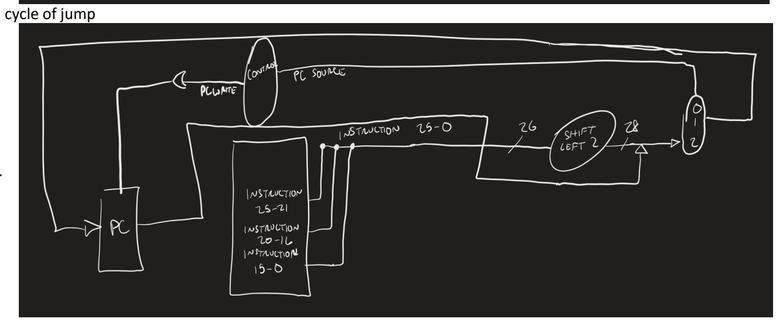
1. CPU – single-cycled implementation
   Consider the following time delays shown in each component (RED colored) and disregard all other times.
   (a) Show the series of component times for all paths for add instruction.
   (b) Show the series of component times for all paths for lw instruction.
   (c) Show the series of component times for all paths for sw instruction.
   (d) Show the series of component times for all paths for beq instruction.
   (e) What will be the system clock cycle time? Answer in ns and justify your answer.
   (f) For a j (jump) instruction, what is the longest path time? Answer in ns and justify your answer.

a. Add
   i. Read address (6ns)
   ii. Add 4 to PC (3ns – occurs simultaneously)
   iii. Read registers (4ns)
   iv. ALU (5ns)
   v. Write data to registers (4ns)
   vi. = **19 ns**
b. Load word
   i. Read address (6ns)
   ii. Add 4 to PC (3ns – occurs simultaneously)
   iii. Read registers (4ns)
   iv. ALU (5ns)
   v. Read data memory (6ns)
   vi. Write registers (4ns)
   vii. = **25 ns**
c. Store word
   i. Read address (6ns)
   ii. Add 4 to PC (3ns – occurs simultaneously)
   iii. Read registers (4ns)
   iv. ALU (5ns)
   v. Write data memory (6ns)
   vi. = **21 ns**
d. Branch if equal
   i. Read address (6ns)
   ii. Add 4 to PC (3ns – occurs simultaneously)
   iii. Read registers (4ns)
   iv. ALU (5ns)
   v. Add ALU (3ns – occurs simultaneously)
   vi. = **15 ns**
e. System clock time equals the slowest data path possible which is
   i. reading the address (6ns), adding 4 to PC (not counted)
   ii. reading the registers (4ns)
   iii. performing ALU operation (5ns)
   iv. performing add ALU (not counted)
   v. read/write to data memory (6ns)
   vi. writing to register (4ns)
   vii. = **25 ns**
f. Jump instruction requires at most
   i. reading the current address (6ns)
   ii. updating the PC with the new jump address (no time)
   iii. = **6 ns**.

2. CPU – multi-cycled implementation
   Consider the following time delays shown in each component (RED colored) and disregard all other times.
   (a) What will be the system clock cycle time? Answer in ns and justify your answer.
   (b) For the following four instruction executions, compute the speedup of using the multi-cycled implementation over the single-cycled implementation.
   add; lw; sw; beq; j(jump);
   (c) Show the datapath and control used in the 3rd cycle of executing a beq instruction. You should draw a subdiagram with only needed parts.
   (d) Show the datapath and control used in the 4th cycle of executing a lw instruction. You should draw a subdiagram with only needed parts.
   (e) Show the datapath and control used in the 3rd cycle of executing a j (jump) instruction. You should draw a subdiagram with only needed parts.

a. The clock cycle time equal to the slowest clock cycle that occurs. Since each cycle only performs one operation on one component. The following is the speed of each components
   i. Memory (6ns)
   ii. Registers (4ns)
   iii. ALU (5ns)
   iv. The clock cycle time is therefore 6ns, the time it takes for the slowest component to occur
b. Speedup of each operation. Each prior instruction will take the time of the longest instruction as that equates to the cycle time. In this case the single cycle implementation takes 25 ns for any one operation, so this will be the old time used in our speedup calculation.
   i. Add
      i. Instruction fetch
      ii. Instruction decode/register fetch
      iii. ALU
      iv. Write data to register
      v. This performs 4 steps which take 6 ns each so the add instruction takes 24 ns. This makes the speedup = 25 ns / 24 ns = 1.042 speedup.
   ii. Load word
      i. Instruction fetch
      ii. Instruction decode/register fetch
      iii. Address computation
      iv. Load MDR
      v. Write data to register
      vi. This performs 5 steps which take 6 ns each so the load word instruction takes 30 ns. This makes the speedup = 25 ns / 30 ns = 0.83 speedup (this instruction is slower but it is the only one).
   iii. Store word
      i. Instruction fetch
      ii. Instruction decode/register fetch
      iii. Address computation
      iv. Write to memory
      v. This performs 4 steps which take 6 ns each so the store word instruction takes 24 ns. This makes the speedup = 25 ns / 24 ns = 1.042 speedup.
   iv. Branch if equal
      i. Instruction fetch
      ii. Instruction decode/register fetch
      iii. Perform logic and set PC if logic passes
      iv. This performs 3 steps which take 6 ns each so the branch if equal instruction takes 18 ns. This makes the speedup = 25 ns / 18 ns = 1.39 speedup.
   v. Jump
      i. Instruction fetch
      ii. Instruction decode/register fetch
      iii. Set PC to new address
      iv. This performs 3 steps which take 6 ns each so the jump instruction takes 18 ns. This makes the speedup = 25 ns / 18 ns = 1.39 speedup.
c. 3rd cycle of branch if equal
   i. 

d. 4th cycle of load word
   i. 

e. 3rd cycle of jump
   i. 

.