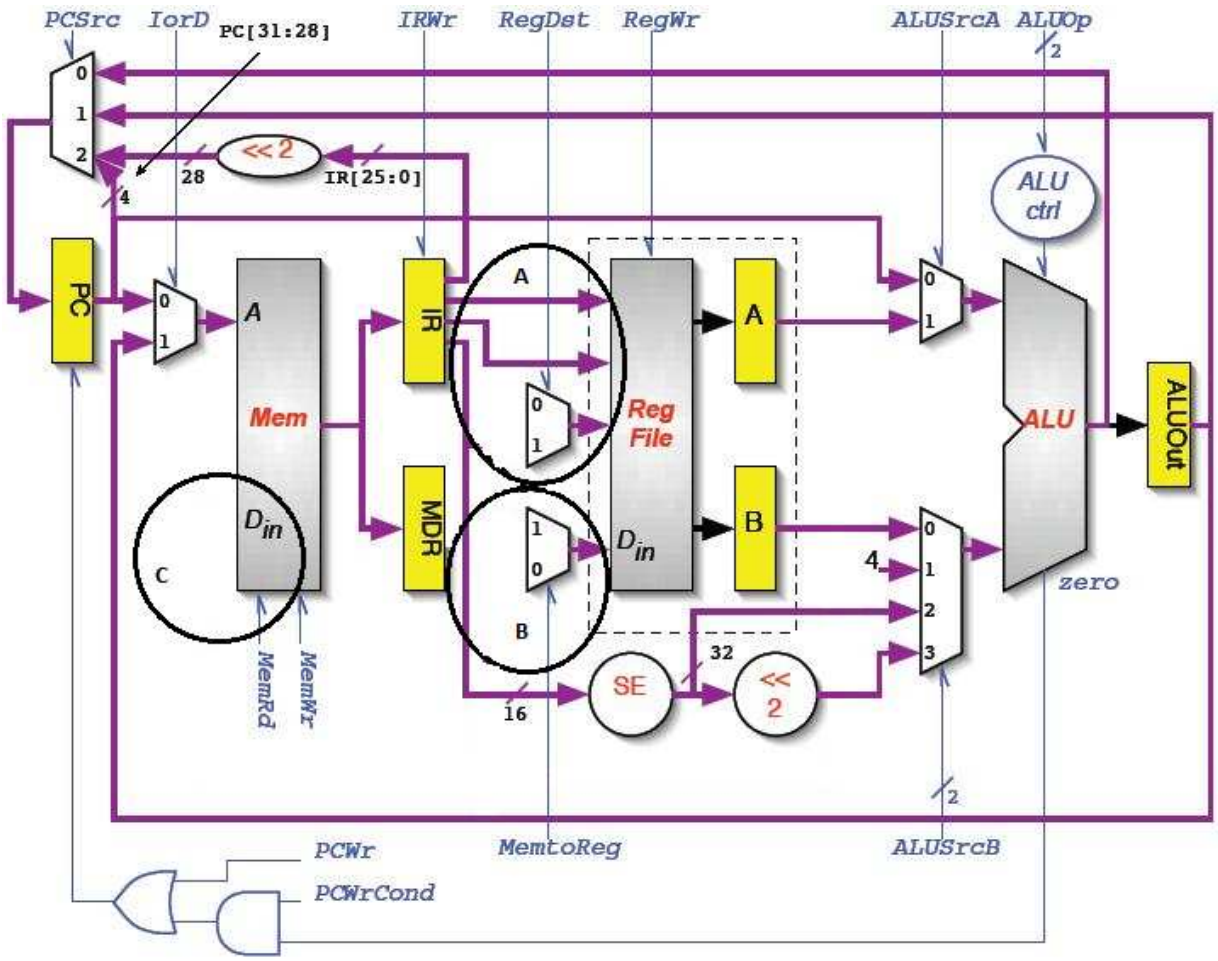
ECE 30, Lab #8 Spring 2014



Shown above is a multi-cycle CPU. There are six special registers in this datapath: `PC`, `IR`, `MDR`, `A`, `B`, and `ALUOut`. Of these, `PC` and `IR` are *enabled to change* when `PCWr` and `IRWr` are high (logic "1") respectively. These control signals change shortly after the sampling (rising) edge of the clock, as the control-FSM changes its state. Therefore, the outputs of these registers do not change until the next sampling edge of the clock after the enabling control signals are asserted.
Additionally, assume that:

- This multi-cycle CPU breaks instructions down into the following stages, each of which takes exactly one clock cycle:

  IF: fetch instruction; compute `PC+4`
  ID/RF: decode instruction; fetch source operands; compute branch target

EX: perform arithmetic/logic operation or compute data memory address
DM: data memory access
WB: write result to register file

- `PC+4` and the *branch target* (`PC+4 + 4*offset`) are computed in the IF and ID/RF stages respectively by the ALU. *Note that these are computed regardless of the instruction type.*

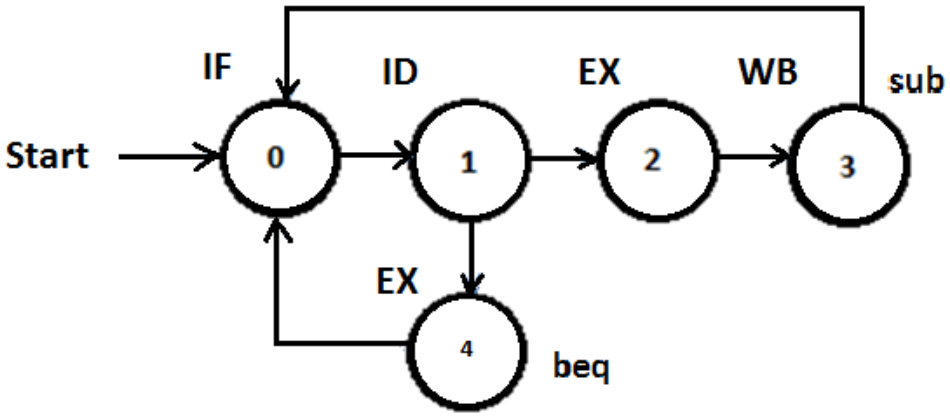
1. Note that there are 5 items/connections missing in the CPU diagram. In this problem you will fill in these five missing items. Each circle (A, B, C) highlights an area missing a single or multiple connection(s)/item(s). For each circle:

   - Draw the missing connection(s)/item(s) onto the CPU diagram.
   - Then, do the following:
     - Briefly explain what you added to the diagram (in case you add a connection, also specify the number of bits the connection represents; if this number is less than 24, then also specify which exact bits are connected from the source of the connection).
     - Give an instruction which uses the added portion of the CPU. Choose from one of the following basic instructions: `j, beq, add/sub, lw, sw`.
     - Give the CPU stage which uses the added portion of the CPU for this instruction (IF, ID/RF, EX, DM, WB).

2. In this problem you will examine the control signals used in executing the `sub` and `beq` instructions. In addition to the information provided before, the following information will be useful for this problem:

   `ALUOp` = 00 signals the ALU to *add* `A` and `B`.
   `ALUOp` = 01 signals the ALU to *subtract* `B` from `A`.
   `ALUOp` = 10 signals the ALU to do the operation specified by the R-format function code.

   Shown below is the state diagram for the `sub` and `beq` instructions.



   Complete the state table shown below for the `sub` and `beq` instructions. Please fill out the table in binary. Use dashes ("-") for unspecified or don't care values. For don't cares: do NOT assign a random value; always clearly indicate it is a don't care.

	0	1	2	3	4
IorD					
MemRd					
MemWr					
IRWr					
ALUSrcA					
ALUSrcB					
ALUOp					
PCSrc					
PCWr					
PCWrCond					
RegWr					
RegDst					
MemtoReg					
	IF	ID/RF	EX	WB	EX

3. (a) Assume the instruction mix shown below for a *multi-cycle* implementation. Complete the table with i) the number of clock cycles ("#CC") and ii) the "Phases" (in the right order) it takes to execute the following instruction types. What is the average CPI of a machine that implements this instruction mix? If the clock period of an alternative *single-cycle* implementation is equal to the time it takes to execute the longest multi-cycle instruction, how much faster is the multi-cycle implementation?

Instruction	Frequency	#CC	Phases (IF, ID/RF, EX, DM, WB)
Arithmetic/Logical	60%		
`lw`	20%		
`sw`	10%		
`beq`	5%		
`j`	5%		

   (b) How many clock cycles would the `jr` instruction take? Explain why it is the same as / different from a `j` instruction.

4. Consider a processor with a **direct-mapped write-through cache with 8 one-word blocks**. Assume that the memory address is 32 bits wide and the memory is byte-addressable.

   (a) Show the layout of the cache, including the data, valid and tag bits, and any logic required to determine hit/miss and select the appropriate data item when reading from the cache. Also, indicate which bits in the 32-bit memory address are used as block offset (if applicable), byte offset, tag, and index, and show where each of these groups of bits are used in the cache architecture. Make sure to label the width of all fields and signals.

   (b) What is the total amount of memory (in bytes) required to build this cache (including both data and other necessary bits)? (show calculations)

   (c) What is the block offset (if applicable), byte offset, tag, and index for word address 56? Give your answer in decimal notation.

(d) Given is the series of address references given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17. Assume the cache is initially empty. Complete the table below. Label each reference as a hit or a miss and show the final contents of the cache in the diagram you drew under (a).

Address	Index	Tag	Hit/Miss
1			
4			
8			
5			
20			
17			
19			
56			
9			
11			
4			
43			
5			
6			
9			
17			

(e) The following miss rate measurements have been made. Instruction miss rate is 6%; data miss rate is 8%. Assume that one-half of the instructions contain a data reference and that the cache miss penalty in no. of clock cycles is (6 + (Block size in words)). Calculate the average miss penalty per instruction.

5. Consider a processor with a **direct-mapped write-through cache with four-word blocks and a total size of 16 words**. Assume that the memory address is 32 bits wide and the memory is byte-addressable.

(a) Show the layout of the cache, including the data, valid and tag bits, and any logic required to determine hit/miss and select the appropriate data item when reading from the cache. Also, indicate which bits in the 32-bit memory address are used as block offset (if applicable), byte offset, tag, and index, and show where each of these groups of bits are used in the cache architecture. Make sure to label the width of all fields and signals.

(b) What is the total amount of memory (in bytes) required to build this cache (including both data and other necessary bits)? (show calculations)

(c) What is the block offset (if applicable), byte offset, tag, and index for word address 56? Give your answer in decimal notation.

(d) Given is the series of address references given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17. Assume the cache is initially empty. Complete the table below. Label each reference as a hit or a miss and show the final contents of the cache in the diagram you drew under (a).

Address	Block Offset	Index	Tag	Hit/Miss
1				
4				
8				
5				
20				
17				
19				
56				
9				
11				
4				
43				
5				
6				
9				
17				

(e) The following miss rate measurements have been made. Instruction miss rate is 4%; data miss rate is 5%. Assume that one-half of the instructions contain a data reference and that the cache miss penalty in no. of clock cycles is (6 + (Block size in words)). Calculate the average miss penalty per instruction.

6. Consider a processor with a **two-way set associative write-back cache with four-word blocks and a total size of 16 words**. Assume that the memory address is 32 bits wide and the memory is byte-addressable.

(a) Show the layout of the cache, including the data, all special bits, and any logic required to determine hit/miss and select the appropriate data item when reading from the cache. Also, indicate which bits in the 32-bit memory address are used as block offset (if applicable), byte offset, tag, and index, and show where each of these groups of bits are used in the cache architecture. Make sure to label the width of all fields and signals.

(b) What is the total amount of memory (in bytes) required to build this cache (including both data and other necessary bits)? (show calculations)

(c) What is the block offset (if applicable), byte offset, tag, and index for word address 56? Give your answer in decimal notation.

(d) Given is the series of address references given as word addresses: 1, 4, 8, 5, 20, 17, 19, 56, 9, 11, 4, 43, 5, 6, 9, 17. Assume the cache is initially empty and LRU is used as replacement policy. Complete the table below. Label each reference as a hit or a miss and show the final contents of the cache in the diagram you drew under (a).

Address	Block offset	Set index	Tag	Set entry	Hit/Miss
1					
4					
8					
5					
20					
17					
19					
56					
9					
11					
4					
43					
5					
6					
9					
17					

(e) The following miss rate measurements have been made. Instruction miss rate is 3%; data miss rate is 4%. Assume that one-half of the instructions contain a data reference and that the cache miss penalty in no. of clock cycles is (6 + (Block size in words)). Calculate the average miss penalty per instruction.