0907731 Advanced Computer Architecture (Fall 2019)
Midterm Exam

رقم التسلسل: .... رقم التسجيل: .... الاسم: ....

==

Instructions: Time **70** min. Open book and notes exam. No electronics. Please answer all problems in the space provided and limit your answer to the space provided. No questions are allowed. There are six problems and each problem has 5 points.

==

P1. What has limited the rate of growth of the clock rate, and what are architects doing with the extra transistors now to increase performance?

The solution:

The increase in power consumption to levels higher than 100 Watts limited the growth of the clock rate as the power consumption is proportional to the clock rate. Cooling such processors is not feasible.

Architects are using extra transistors to have more cores, larger caches, and more integration.

P2. You are designing a system for a real-time application in which specific deadlines must be met. Finishing the computation faster gains nothing. You find that your system can execute the necessary code, in the worst case, twice as fast as necessary.

a) How much energy do you save if you execute at the current speed and turn off the system when the computation is complete?

The solution:

Assume $P_{Base} = ½ C V^2 f$

$\text{Energy}_{Base} = \text{Time} \times P_{Base} = T \times (½ C V^2 f)$

$\text{Energy}_{a} = \text{Time}_{a} \times P_{Base} = (½ T) \times (½ C V^2 f)$

Saved energy $= (1 - (\text{Energy}_{a} / \text{Energy}_{Base})) \times 100\% = 50\%$

b) Assuming that the main energy consumption is due to dynamic power, how much energy do you save if you set the voltage and frequency to be half as much?

The solution:

$\text{Energy}_{b} = \text{Time} \times P_{b} = T \times (½ C (½V)^2 (½f))$

Saved energy $= (1 - (\text{Energy}_{b} / \text{Energy}_{Base})) \times 100\%$

$= (1 – (1/8)) \times 100\% = 87.5\%$

P3. Showing all bus widths, draw a four-way associative cache with the following specifications: address width = 12 bits, number of sets = 16, block size = 128 bits, word size = 4 bytes, and write through scheme.

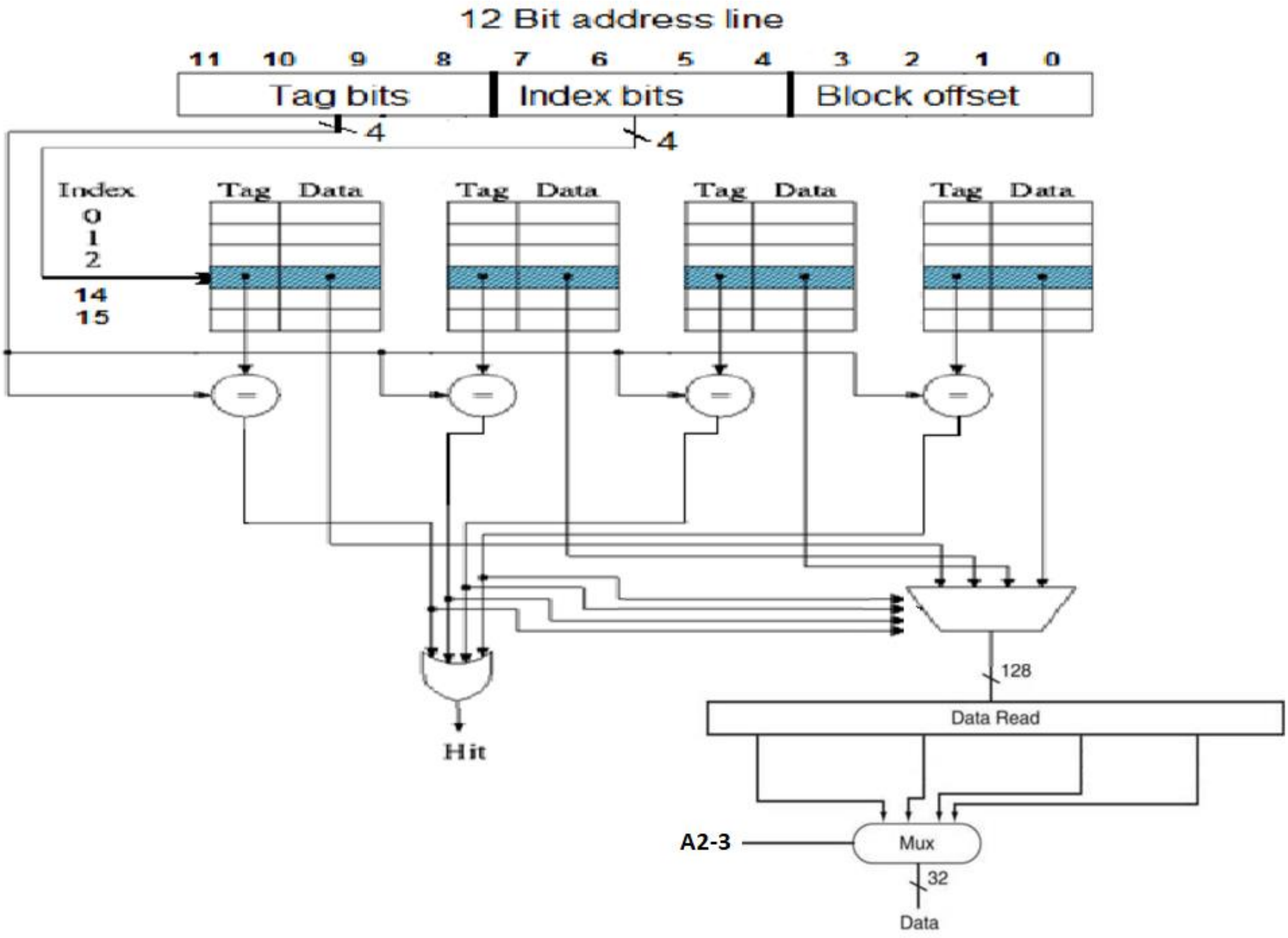
The solution is:

<block offset> = $\lg_2$ (block size in bytes) = $\lg_2$ (128/8) = 4 bits

<index> = $\lg_2$ (No. of sets) = $\lg_2$ (16) = 4 bits

<tag> = 12 - <index> - <block offset> = 12 – 4 – 4 = 4 bits

The valid bit is not drawn below.



P4. A processor runs on a 2-GHz clock and has a Level 1 cache of 1-cylce hit time and 5% miss rate. The miss penalty to the memory is 25 ns. You are considering adding a Level 2 cache to achieve an average memory access time of 1.4 cycles. Which cache configuration out of the options shown in the table below you select to achieve this AMAT?

Size	**64 KB**	**128 KB**	**256 KB**	**512 KB**	**1 MB**
Hit time	3.0 ns	3.0 ns	3.0 ns	3.5 ns	3.5 ns
Miss rate	5.0%	4.5%	4.0%	3.8%	3.7%

The solution:

T = 1 / f = 1 / 2 GHz = 0.5 ns

25 ns / 0.5 ns = 50 cycles

L1:

1.4 = 1 + 0.05 × Miss Penalty

Miss Penalty = 0.4 / 0.05 = 8 cycles

L2:

8 = Hit Time (L2) + Miss Rate (L2) × 50

A cache of size 256 KB satisfies the above equation

8 = 3 ns × 2 GHz + 4% × 50

8 = 6 + 2

P5. Unroll the following loop two times and schedule it for the 5-stage pipeline studied in the class. Assume that the pipeline has all necessary forwarding paths and branch instructions are resolved in the decode stage. For simplicity, assume that the loop is executed an even number of iterations.

```
Loop:    lw     x1,0(x2)
         addi   x1,x1,1
         sw     x1,0(x2)
         addi   x2,x2,4
         sub    x4,x3,x2
         bnz    x4,Loop
```

The solution:

1) **Replicate and modify and rename:**

```
Loop: lw    x1,0(x2)
      addi  x1,x1,1
      sw    x1,0(x2)
      lw    x10,4(x2)
      addi  x10,x10,1
      sw    x10,4(x2)
      addi  x2,x2,8
      sub   x4,x3,x2
      bnz   x4,Loop
```

2) **Schedule:**

```
Loop: lw    x1,0(x2)
      lw    x10,4(x2)
      addi  x1,x1,1
      sw    x1,0(x2)
      addi  x10,x10,1
      addi  x2,x2,8
      sub   x4,x3,x2
      sw    x10,-4(x2)
      bnz   x4,Loop
```

P6. Assume that the following code sequence is executed by a speculative pipelined processor. This processor uses reservation stations, common data bus, and reorder buffer. All stages other than FP execution take one cycle each. Floating-point addition takes 2 cycles. The processor has one address calculation unit, one memory access unit, one integer ALU unit, one branch unit, and one FP unit. Using the multi-cycle pipeline diagram below, specify the execution of these instructions in this processor pipeline. Assume that the branch is correctly predicted as a not taken branch.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
`ld      x2,0(x1)`	***F***	***I***	***A***	***M***	***W***	***C***											
`fld     f3,0(x2)`		***F***	***I***			***A***	***M***	***W***	***C***								
`fadd.d f4,f5,f3`			***F***	***I***					***E***	***E***	***W***	***C***					
`beq     x2,x3,8`				***F***	***I***	***E***	***W***						***C***				
`sd      x2,0(x10)`					***F***	***I***	***A***							***C***			
`ld      x6,0(x10)`						***F***	***I***	***A***							***M***	***W***	***C***

<Good Luck>