0907731 Advanced Computer Architecture (Spring 2019)
Midterm Exam

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

==

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. In 1975, Gordon Moore revised his law forecasting that the number of transistors in a dense integrated circuit to double about every two years. The largest IC at that time had about 10,000 transistors. What is the forecasted maximum number of transistors on a chip for the Year 1985?

The solution:

Number of transistors $= N_0 \times 2^{((1985-1975)/2)}$

$= 10{,}000 \times 2^5$

$= 10{,}000 \times 32$

$= 320{,}000$ **transistors**

P2. Assume that you have a program that executes 10^9 instructions. You have the option of running it on a clock of 2.4 GHz with CPI=1.2 or on 1.0 GHz clock with CPI=1.0.

a) Which option is faster and by how much?

The solution:

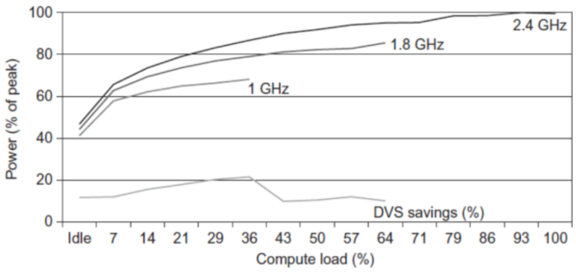
CPU Time = Instruction Count × CPI / Frequency

Option 1: Time $= 10^9 \times 1.2 / (2.4\times10^9) = 0.5$ **seconds**

Option 2: Time $= 10^9 \times 1.0 / (1.0\times10^9) = 1.0$ **seconds**

Option 1 is faster by (1.0 / 0.5 - 1) × 100% = 100%

b) Given the relative power consumption shown in the following chart and assuming peak compute load for the two options, which option is more energy efficient and by how much?



The solution:

Energy = Power × Time

Option 1: Relative Energy = 100% × 0.5 = 0.5

Option 2: Relative Energy = 65% × 1.0 = 0.65

Option 1 is more efficient by (0.65 / 0.5 - 1) × 100% = 30%

P3. Showing all bus widths, draw a two-way associative cache with the following specifications: cache size = 128 KB, block size = 32 bytes, word size = 4 bytes, address width = 64 bits, and write through scheme.

The solution is:

m = 64

n = $\lg_2$ (block size in bits) = $\lg_2$ (32 × 8 bits) = 8 bits

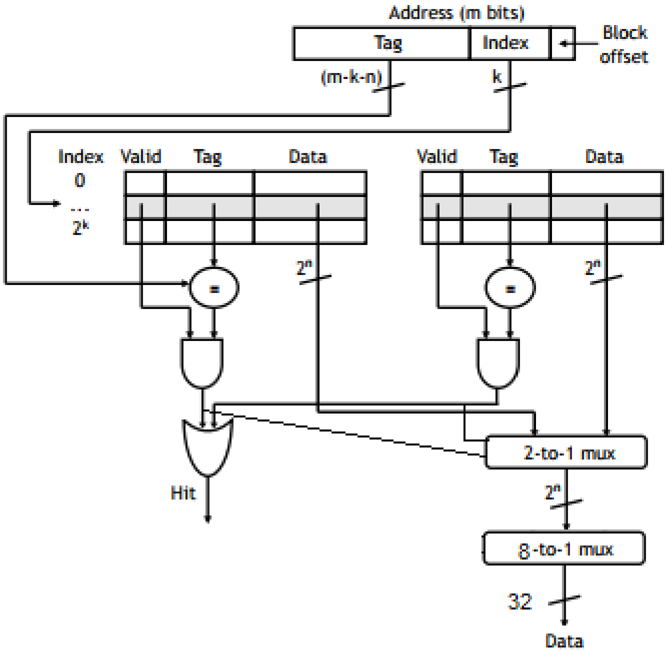
<block offset> = $\lg_2$ (block size in bytes) = $\lg_2$ (32) = 5 bits

Number of blocks = 128 KB / 32 bytes = 4 K blocks

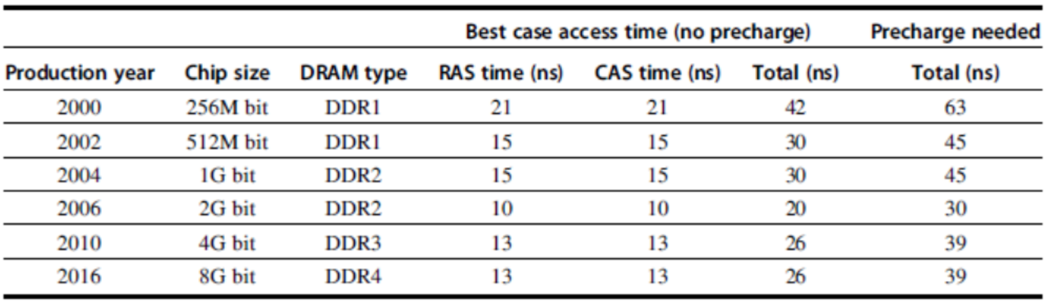
Number of sets = 4 K / 2 = 2 K sets

k = <index> = $\lg_2$ (No. of sets) = $\lg_2$ (2 K) = 11 bits

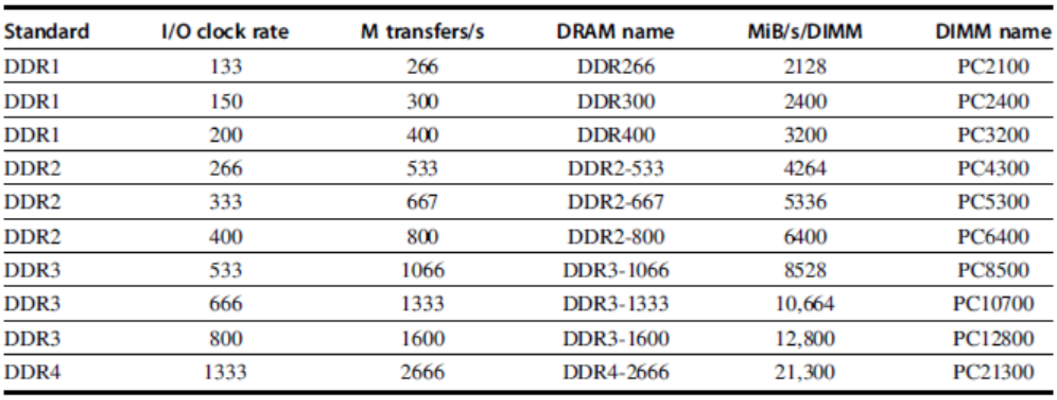
<tag> = 64 - <index> - <block offset> = 64 – 11 – 5 = 48 bits



P4. Consider the following two tables that show the specifications of DDR4 and PC21300.



			Best case access time (no precharge)			Precharge needed
Production year	Chip size	DRAM type	RAS time (ns)	CAS time (ns)	Total (ns)	Total (ns)
2000	256M bit	DDR1	21	21	42	63
2002	512M bit	DDR1	15	15	30	45
2004	1G bit	DDR2	15	15	30	45
2006	2G bit	DDR2	10	10	20	30
2010	4G bit	DDR3	13	13	26	39
2016	8G bit	DDR4	13	13	26	39



Standard	I/O clock rate	M transfers/s	DRAM name	MiB/s/DIMM	DIMM name
DDR1	133	266	DDR266	2128	PC2100
DDR1	150	300	DDR300	2400	PC2400
DDR1	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1333	2666	DDR4-2666	21,300	PC21300

a) What is the data width of PC21300?

The solution:

Data Width = Bandwidth / Transfer rate

= 21,300 MiB/s / 2666 M transfers/s = 8 bytes

b) What is the total time needed to read a block of 128 bytes from this module?

The solution:

Time = RAS time + CAS time + Transfer time

= 13 ns + 13 ns + (128 / 21300 MiB/s) × 10^9

= 26 + 128,000 / 21300 ns

= 26 + 6

= 32 ns

P5. Assume the latencies shown in the following table.

Instruction producing result	Instruction using result	Latency in clock cycles
FP ALU op	Another FP ALU op	3
FP ALU op	Store double	2
Load double	FP ALU op	1
Load double	Store double	0

Unroll the following loop two times and use the table below to schedule the unrolled loop efficiently for a VLIW processor that has one memory reference, one FP operation, and one integer/branch operation fields.

```
Loop:   fld     f31,0(x20)      // f31=array element
        fadd.d  f31,f31,f21     // add scalar in f21
        fsd     f31,0(x20)      // store result
        addi    x20,x20,-8      // decrement pointer
        blt     x22,x20,Loop    // branch if x22 < x20
```

Memory reference	FP operation	Integer/branch operation
`fld   f31,0(x20)`		
`fld   f30,-8(x20)`		
	`fadd.d   f31,f31,f21`	
	`fadd.d   f30,f30,f21`	
		`addi  x20,x20,-16`
`fsd   f31,16(x20)`		
`fsd   f30,8(x20)`		`blt   x22,x20,Loop`

P6. Assume that the following code sequence is executed by a double-issue speculative pipelined processor. This processor uses reservation stations, common data buses, and reorder buffer. All stages other than FP execution take one cycle each. Floating-point addition takes 4 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 incorrectly 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***	***E***	***E***	***W***	***C***			
`beq     x2,x3,8`		***F***	***I***			***E***	***W***							***C***			
`fsd     f4,0(x10)`			***F***	***I***	***A***									***n***			
`ld      x6,12(x1)`			***F***	***I***			***A***	***M***	***W***					***n***			

<Good Luck>