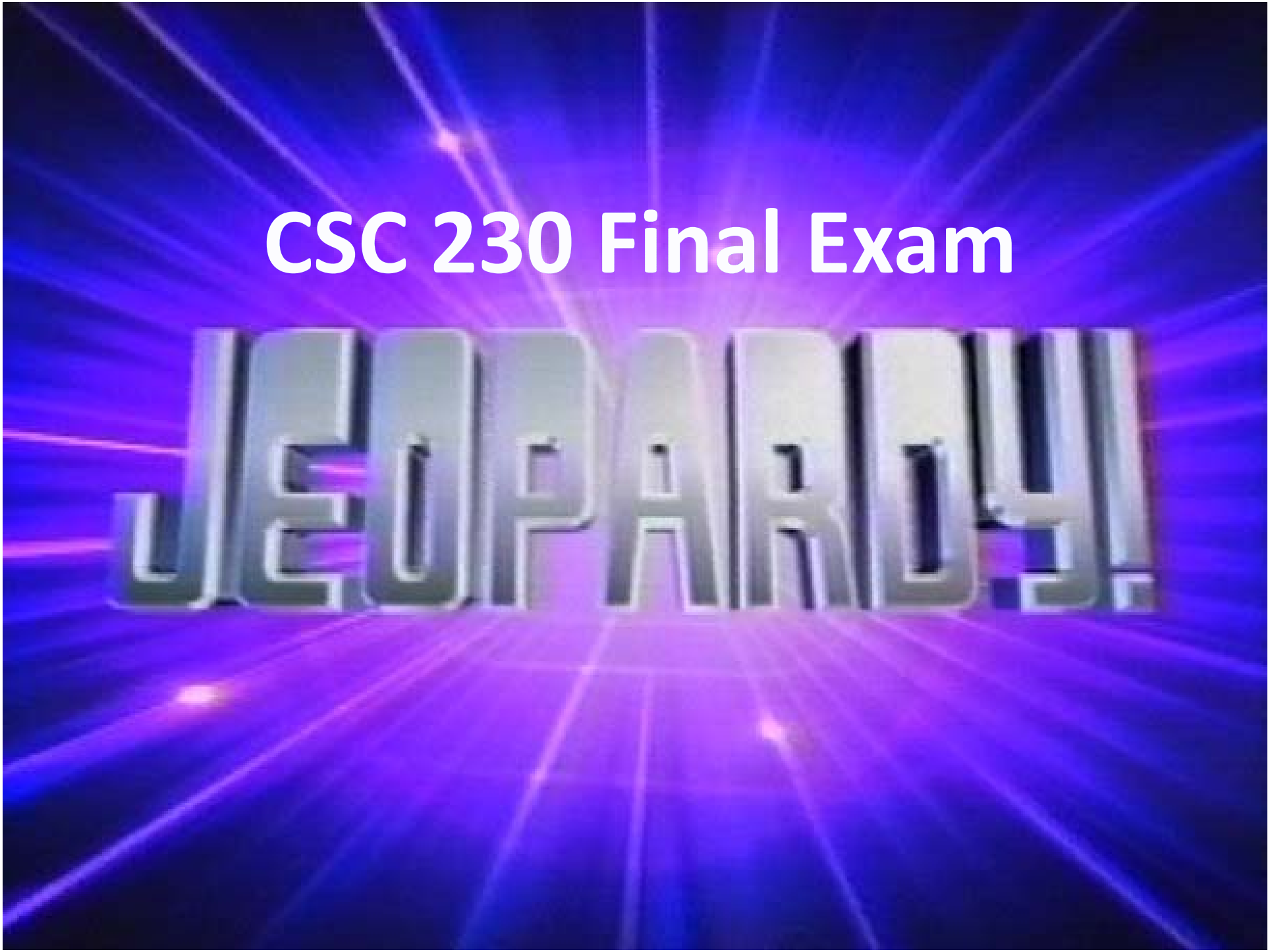
CSC 230 Final Exam



CSC 230 Jeopardy

What's the difference?	Pipelines	Fetch! Decode! Execute!	Addressing Modes	Virtual Memory	Mish mash math
100	100	100	100	100	100
200	200	200	200	200	200
300	300	300	300	300	300
400	400	400	400	400	400
500	500	500	500	500	500

T1 - 100

- What is the difference between volatile and non-volatile memory? (What is an example or application of non-volatile memory?)

T1 - 100

- What is the difference between volatile and non-volatile memory? (What is an example or application of non-volatile memory?)

Non-volatile memory keeps it contents when the power is turned off.
Example: flash drive

T1 - 200

- What are two differences between L1 (primary) and L2 (secondary) cache?

T1 - 200

- What are two differences between L1 (primary) and L2 (secondary) cache?

•the primary cache of a multilevel cache is often smaller
•the primary cache often uses a smaller block size
•the primary cache tries to optimize the hit time to have a shorter clock cycle

•the secondary cache usually larger, access time less critical
•the secondary cache often uses larger block size
•the secondary cache optimizes the miss rate to reduce the penalty of long memory access time

T1 - 300

- What are two differences between a subroutine and an interrupt service routine?

T1 - 300

- What are two differences between a subroutine and an interrupt service routine?

A1. Subroutine performs tasks required by control flow of program
A2. Interrupt is not scheduled by the program, may even be caused by or related to other processes/users

B1. Subroutine always returns to where it was called from and program resumes execution
B2. After an interrupt, execution may:
Return where it left off (as in a subroutine call);
Return to a different place in the application program (e.g. like a 'catch' block in Java)
Exit from the application and return to the OS;
Exit from the OS and shut down (the 'blue screen of death'?).

T1 - 400

- Access to data on disk is much slower than to data in main memory. What are the factors in computing disk access time?

T1 - 400

- Access to data on disk is much slower than to data in main memory. What are the factors in computing disk access time?

SEEK TIME + ROTATIONAL DELAY = ACCESS TIME

T1 - 500

- What are 3 of the differences between Static RAM and dynamic. Which would you use for main memory? Which would you use for cache?

T1 - 500

- What are 3 of the differences between Static RAM and dynamic. Which would you use for main memory? Which would you use for cache?

Dynamic RAM is
- More dense
- Less expensive
- Slower
- Needs refreshing
- Simpler

STATIC RAM: when speed is needed, e.g. Cache
DYNAMIC RAM: density desired, e.g. main memory

T2 - 100

Time →

\ Clock \cycle Instruction	1	2	3	4	5	6
I1	F1	D1	E1	W1		
I2		F2	D2	E2	W2	
I3			F3	D3	E3	W3
I4				F4	D4	E4
I5					F5	D5

The example above shows the sequence of 5 instructions in a 4 stage pipeline. Which time slots are the most efficient/effective in terms of CPU usage? Why?

T2 - 100

Time →

\ Clock \cycle Instruction	1	2	3	4	5	6
I1	F1	D1	E1	W1		
I2		F2	D2	E2	W2	
I3			F3	D3	E3	W3
I4				F4	D4	E4
I5					F5	D5

The example above shows the sequence of 5 instructions in a 4 stage pipeline. Which time slots are the most efficient/effective in terms of CPU usage? Why?

4 and 5.

T2 - 200

- What sort of pipelining hazard is in the following code? Why?

```
[1]     MUL     r1,r2,r3
[2]     LDR     r1,[r1]
```

T2 - 200

- What sort of pipelining hazard is in the following code? Why?

```
[1]     MUL     r1,r2,r3
[2]     LDR     r1,[r1]
```

Data Hazard - The source r1 in line [2] is the destination in the line above.

T2 - 300

- Define the speedup which could possibly be obtained, in the best case, by using a pipeline with m stages executing N instructions.

T2 - 300

- Define the speedup which could possibly be obtained, in the best case, by using a pipeline with m stages executing N instructions.

~m

T2 - 400

- **What sort of pipelining hazard is in the following code? Why?**

```
     CMP   r1, #5
     BEQ   B1          @taken!!!
     STR   r1, [r5]
B1:  STR   r1, [r4]
```

T2 - 400

- **What sort of pipelining hazard is in the following code? Why?**

```
     CMP   r1, #5
     BEQ   B1          @taken!!!
     STR   r1, [r5]
B1:  STR   r1, [r4]
```

Instruction Hazard.
Branched to B1. This instruction has not been fetched yet.

T2 - 500

- What is the penalty incurred from a branch in a system using a pipeline with m stages executing N instructions?

T2 - 500

- What is the penalty incurred from a branch in a system using a pipeline with m stages executing N instructions?

m-1

T3 - 100

- In the following instruction, R1 gets the number 12 from the datapath.
- What part of the processor does the number #12 come from?

?? → datapath → R1

T3 - 100

- In the following instruction, R1 gets the number 12 from the datapath.
- What part of the processor does the number #12 come from?

?? → datapath → R1

IR = The instruction register

T3 - 200

- In the instruction

  LDR R2,[R5]

- What are the two parts missing from the execute phase below?

  R5 --> datapath --> **??**

  Read Signal --> Control line

  wait for memory to respond

  ?? --> datapath --> R2

T3 - 200

- In the instruction

  LDR R2,[R5]

- What are the two parts missing from the execute phase below?

  R5 --> datapath --> **??**

  Read Signal --> Control line

  wait for memory to respond

  ?? --> datapath --> R2

1. MAR (Memory Address Register) and Address Bus
2. Data Bus and MDR (Memory Data Register)

Ste
1.
2.
3.
4.
5.

T3 - 300

- What are the 10 steps in a fetch?

Step	Action	Step	Action
1.	PC → MAR	6.	Add Signal → ALU CTRL
2.	MAR → Address Bus	7.	ALU Output → PC
3.	Read Signal → CTRL	8.	Wait for mem read to finish
4.	PC → ALU	9.	Data bus → MDR
5.	#4 → ALU input	10.	MDR → IR

T3 - 400

- What are the 3 steps in the execution phase of this instruction?

  STR R3,[R4]

T3 - 400

- What are the 3 steps in the execution phase of this instruction?

  STR R3,[R4]

R4 → datapath → MAR & Address Bus
R3 → datapath → MDR & Data Bus
Write signal → Control

T3 - 500

- Give the detailed steps performed during the phases which follow Fetch for the instruction:

  adds r0, r1, r2, LSL #2

T3 - 500

- Give the detailed steps performed during the phases which follow Fetch for the instruction:

  adds r0, r1, r2, LSL #2

R2 → datapath → ALU (shifter)
#2 (from IR) → datapath → ALU (shifter)
Shift done
R1 → ALU
ALU output → datapath → R0 and Condition Register set appropriately

T4 - 100

- What are the addressing modes of each of the operands in the instruction below?

  LDR R0,#myData

- What does this instruction do?

T4 - 100

- What are the addressing modes of each of the operands in the instruction below?

  LDR R0,=myData

- What does this instruction do?

R1 – Register
=myData – Pc Relative

R0 has the address of myData

T4 - 200

- What are the addressing modes of each of the operands in the instruction below?

  LDR R0, #3

- What does this instruction do?

T4 - 200

- What are the addressing modes of each of the operands in the instruction below?

  LDR R0, #3

- What does this instruction do?

R1 – Register
#3 – Immediate

R0 = 3

T4 - 300

- What are the addressing modes of each of the operands in the instruction below?

  CMP R0, R1

- What does this instruction do?

T4 - 300

- What are the addressing modes of each of the operands in the instruction below?

  CMP R0, R1

- What does this instruction do?

R0 – Register
R1 – Register

The CPSR is updated with the results of R0 – R1.

T4 - 400

- What are the addressing modes of each of the operands in the instruction below?

  LDR R1,[R2],R3

- What does this instruction do?

T4 - 400

- What are the addressing modes of each of the operands in the instruction below?

  LDR R1,[R2],R3

- What does this instruction do?

R1 – Register
Register Indexed with register offset post-indexed

R1 is assigned the value from memory whose address is in R2.
Then R2 = R2 + R3

T4 - 500

- What are the addressing modes of each of the operands in the instruction below?

STR R1,[R2,R3,LSL #2]!

- What does this instruction do?

T4 - 500

- What are the addressing modes of each of the operands in the instruction below?

STR R1,[R2,R3,LSL #2]!

- What does this instruction do?

R1 – Register
[R2, R3, LSL #2]! - Register Indexed with register offset pre-indexed, with write back and shift

R2 = R2 + R3*4
R1 is assigned the value from memory whose address is in R2.

T5 - 100

- In a computer which has a 64-bit address space with byte addressable memory and pages that are 512 KB in size, how many entries does the page table have?

T5 - 100

- In a computer which has a 64-bit address space with byte addressable memory and pages that are 512 KB in size, how many entries does the page table have?

Each page is 512 KB = 2^9 x 2^{10} = 2^{19}
Total address space = 2^{64} bytes
2^{64} / 2^{19} = 2^{45} pages

T5 - 200

- How many bits are necessary to represent the page table numbers in a virtual address if there are 2^{32} pages in the page table? How many hexadecimal digits does this require?

T5 - 200

- How many bits are necessary to represent the page table numbers in a virtual address if there are 2^{32} pages in the page table? How many hexadecimal digits does this require?

32 bits
8 Hex digits

T5 - 300

- Given this page table and a virtual address where the first 2 nibbles are the page number, what is the physical address for the virtual address 011234

0	08
1	77
2	-
3	14

T5 - 300

- Given this page table and a virtual address where the first 2 nibbles are the page number, what is the physical address for the virtual address 011234

0	08
1	77
2	-
3	14

771234

T5 - 400

- Given this page table and a virtual address where the first 2 nibbles are the page number, what is the physical address for the virtual address 021234

0	08
1	77
2	-
3	14

T5 - 400

- Given this page table and a virtual address where the first 2 nibbles are the page number, what is the physical address for the virtual address 021234

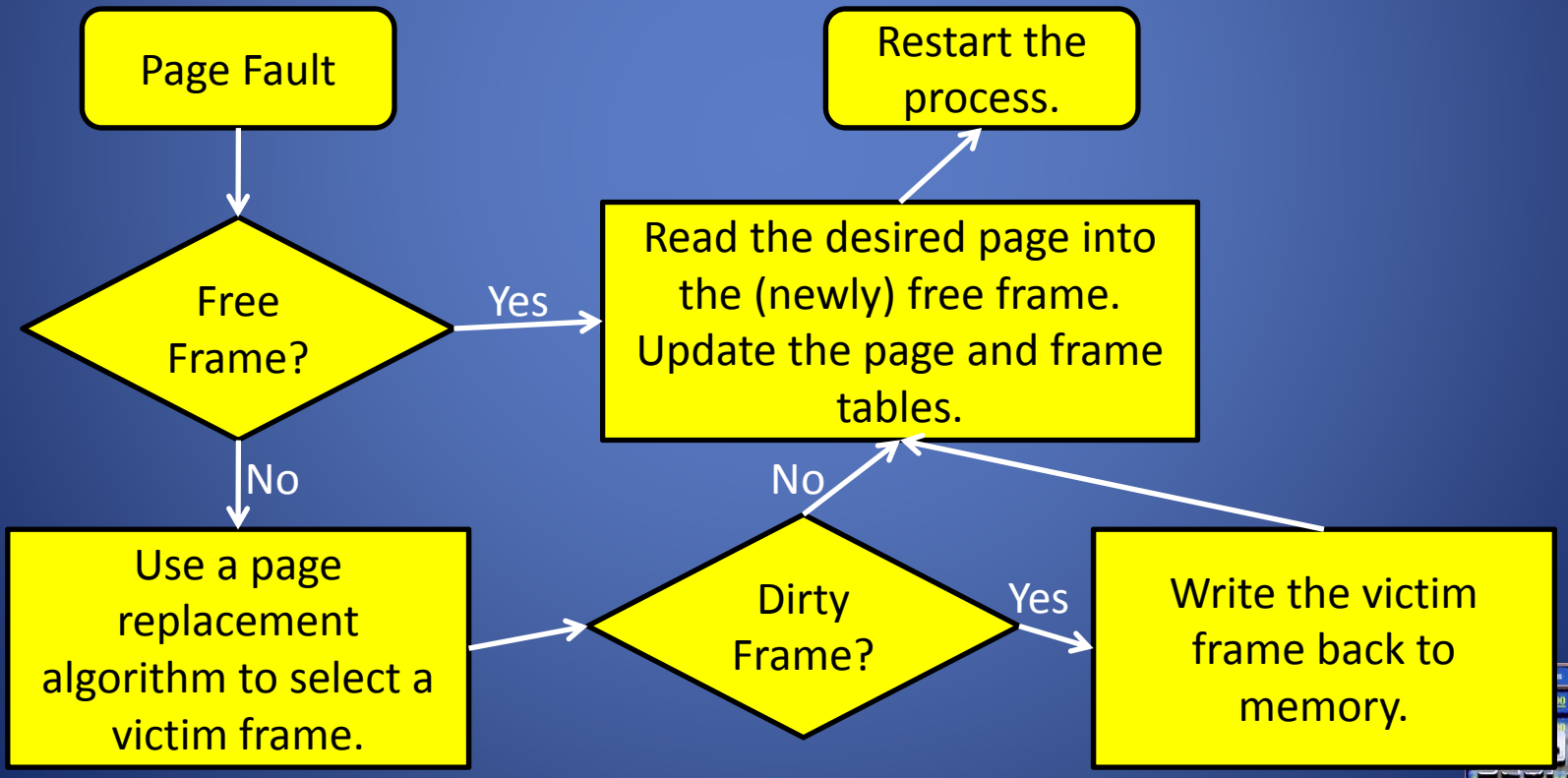
0	08
1	77
2	-
3	14

Page Fault


DAILY
DOUBLE

T5 - 500

- Draw a flow chart of all the steps the Operating System goes through when a page fault occures



T6 - 100

What do f and P represent in Amdhal's law

T6 - 100

What do f and P represent in Amdhal's law

P is the number of processors
f is the fraction f of the basic operations must be performed sequentially

T6 - 200

- Suppose that execution time for a program is directly proportional only to instruction access time.
- Access time to an instruction is 1 ns from the cache and 9 ns from memory. The probability of a cache hit is 99%. In the case of a cache miss, the instruction is fetched from main memory and copied into the cache, and then a second access must take place to copy the instruction from the cache (this time it will be a hit).

- Compute the execution time of a program with 100 instructions with the cache.

T6 - 200

- Suppose that execution time for a program is directly proportional only to instruction access time.
- Access time to an instruction is 1 ns from the cache and 9 ns from memory. The probability of a cache hit is 99%. In the case of a cache miss, the instruction is fetched from main memory and copied into the cache, and then a second access must take place to copy the instruction from the cache (this time it will be a hit).

- Compute the execution time of a program with 100 instructions with the cache.

1*(9ns+1ns) + 99*1ns = 109ns

T6 - 300

- Suppose that we have a DRAM chip which has 1000 rows of 16 bytes each, and that each row can be refreshed in 4 clock cycles
- How many clock cycles are needed to refresh the entire DRAM chip?

T6 - 300

- Suppose that we have a DRAM chip which has 1000 rows of 16 bytes each, and that each row can be refreshed in 4 clock cycles
- How many clock cycles are needed to refresh the entire DRAM chip?

1000 * 4 = 4000 cycles

T6 - 400

- If you have 2^{30} available addresses and disk drives that require 512M of address space each, how many can you fit into the system?

T6 - 400

- If you have 2^{30} available addresses and disk drives that require 512M of address space each, how many can you fit into the system?

$2^{30} / 2^{19} = 2^{11}$

T6 - 500

- If you have 2^{30} available addresses and ½ is used for RAM and ¼ is used for ROM, how much is left for peripherals?

T6 - 500

- If you have 2^{30} available addresses and ½ is used for RAM and ¼ is used for ROM, how much is left for peripherals?