

Final Exam

Name:

SSN (last 4 digits):

Total is 100 points.

You must show all of your work -- partial credit may be given to partially correct answers, while answers with no justification may not receive full points. Use the back of the exam sheets if you need extra space.

Any instances of cheating or copying found during the exam or during grading will be severely dealt with. You will have to leave the class if found engaging in any form of academic dishonesty during the exam.

This exam is x pages long and has n questions.

1) (x pts) Define several characteristics of a load-store architecture.

2) (x pts) “True” performance is measured using a ‘wall clock’. Identify the elements of performance when computing it for a given architecture.

3) (x pts) Name 3 addressing modes used to access data.

4) (x pts) Define PC relative addressing mode, i.e., how is the address computed?

5) (x pts) What are the 'condition codes' produced by an ALU used for?

6) (x pts) Distinguish between a program and a process (in the operating system context), i.e., how are they different?

7) (x pts) Describe the primary disadvantage of a variable-sized memory partitioning scheme that might be used by the OS.

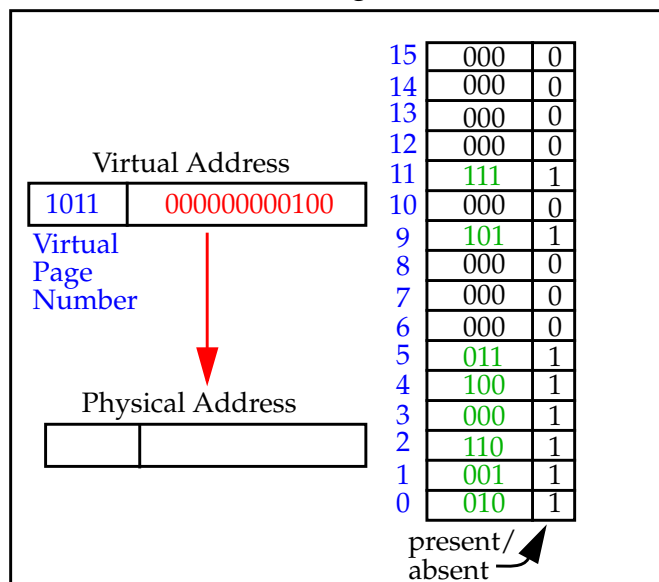
8) (x pts) Define the virtual memory space of a process. How is it different than physical memory?

9) (x pts) What value does segmentation provide to the OS, i.e., how does it help the OS?

10) (x pts) What value does paging provide to the OS, i.e., how does it help the OS?

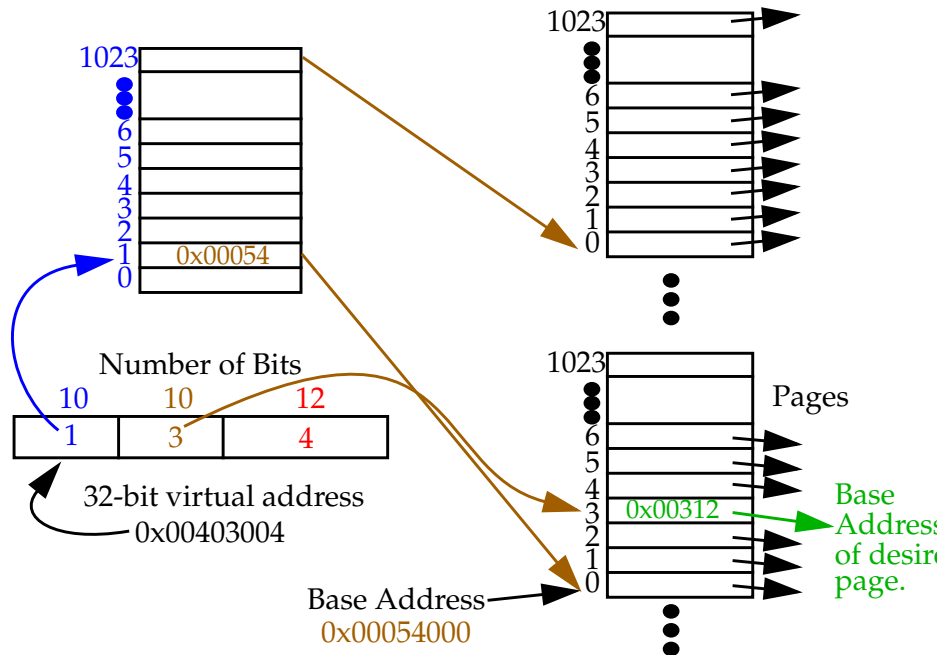
11) (x pts) What is a 'page fault' and how is it handled by the OS?

12) (x pts) Give the physical address for the following virtual address.



13) (x pts) How much overhead is associated with the page tables assuming 4 bytes per page table entry and a 32-bit virtual address space?

14) (x pts) What type of paging scheme is shown in the figure?



15) (x pts) Define TLB (spell it out). What is it used for?

16) (x pts) How is a physical address constructed in 'real mode' on the 8086?

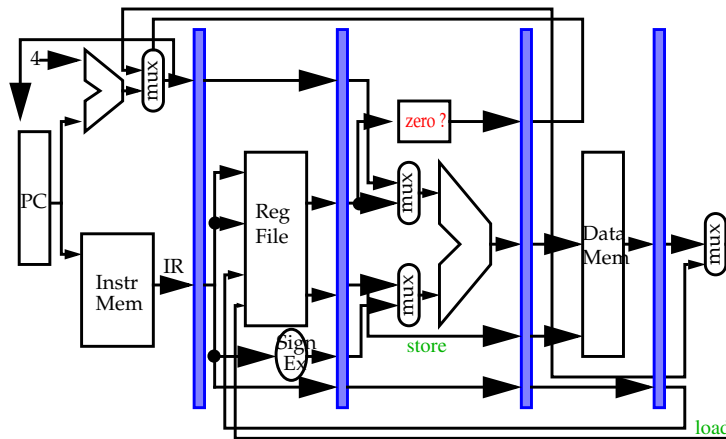
17) (x pts) How are the segment registers in the x86 architectures used in protected mode? I.e., describe how they are used to construct a physical address.

18) (x pts) Name (do not describe) the 5 stages of the DLX pipeline.

19) (x pts) What feature of the unpipelined version of the DLX regarding 'clock-cycles-per-instruction' can NOT be leveraged by the pipelined version.

20) (x pts) What timing constraint limits the expansion of the number of pipeline stages for a given computer architecture to a large number, e.g., 500?

21) (x pts) Briefly describe the problem associated with the following pipelined version of DLX in dealing with branch instructions.



22) (x pts) Define a structural hazard.

23) (x pts) Why does the version of the DLX shown above stall on the following sequence of instructions?

```

ADD R1, R2, R3
SUB R4, R5, R1
AND R6, R1, R7
OR R8, R1, R9
XOR R10, R1, R11
    
```

24) (x pts) What do more advanced versions of the DLX add to prevent these types of stall?

25) (x pts) What specific instruction sequence on the DLX introduces a pipeline 'bubble' even when the DLX is upgraded with the hardware described in question 24?

26) (x pts) How can the compiler help with the problem described in question 25?

27) (x pts) Define a control hazard.

28) (x pts) How many stall cycles do control hazards introduce in the original DLX pipeline?

29) (x pts) What needs to happen in order to avoid control hazards in the DLX pipeline?

30) (x pts) Briefly describe 'delayed branch' as a static prediction scheme used by the compiler to help prevent stalls due to control hazards.