CSE 420 Fundamentals of Computer Architecture
Term: Fall 2026

Self-Assessment

Assigned Aug. 20th, 2026 ***Due Aug. 27, 2026***

https://ascslab.org/courses/CSE420/index.html

This problem set is intended as an assessment to determine your background. For each question, we ask that you fill out the table at the end of the problem set handout indicating your level of confidence with each assigned problem and hand this in with your solutions. If you have never seen the material before, then please enter "0". If you have seen the material, and think you should know it, but can't answer the question without spending time studying your old notes, then please enter "1". If you are comfortable with the material, then enter "2". You should turn in solutions for problems where you entered "1" and "2", but do not have to turn in solutions for problems for which you entered a "0".

If there are too many "0"s in the table then it is unlikely you will be able to keep up with CSE 420 this term, as the pace is such that you will not be able to take prerequisite material concurrently. If you are not sure whether you have the background to take this class, please feel free to discuss your particular situation with the instructors.

You must work individually and turn in your own solutions. Do not discuss the problems with others.

Name:

Part I – CSE 120: Digital Design Fundamentals

Problem 1

Number Systems and Boolean Algebra

A) Convert the decimal number 183 to binary and to hexadecimal.

B) Simplify the Boolean expression F = A·B + A·B'·C + A·C' using Boolean algebra or a Karnaugh map, and give the minimal sum-of-products form.

Problem 2

Combinational Logic

A) Construct the truth table for a 2-bit comparator that outputs 1 when its two 2-bit inputs A and B are equal, and 0 otherwise.

B) Describe, in words or with a gate-level sketch, the logic gates needed to implement F = (A + B)·C'.

Problem 3

Sequential Logic

A) Explain the difference between a D flip-flop and a D latch, and state which one is used to build synchronous sequential circuits.

B) Describe how a 3-bit synchronous up-counter can be built from D flip-flops. What additional logic is needed to make it count up modulo 6 instead of modulo 8?

Problem 4

Verilog Basics

A) Write a Verilog module for a 2-to-1 multiplexer with inputs a, b, sel and output out.

B) What is the difference between blocking (=) and non-blocking (<=) assignment in Verilog, and when should each be used?

Part II – CSE 230: Computer Organization and Assembly Language Programming

Problem 5

Instruction Set Architectures

A) What is the difference between a RISC and a CISC instruction set architecture? Give one example ISA of each.

B) List and briefly describe three common addressing modes (e.g., immediate, register, base/displacement).

Problem 6

Assembly Programming

A) Write RISC-V (or MIPS) assembly code that computes the sum of the integers 1 through 10 and stores the result in a register.

B) Translate the following C code into assembly of your choice: if (a > b) { c = a; } else { c = b; }

Problem 7

Memory Organization

A) Explain the difference between big-endian and little-endian byte ordering. Show how the 32-bit value 0x12345678 would be stored in memory starting at address 0x1000 under each scheme.

Problem 8

I/O Programming

A) What is the difference between polling and interrupt-driven I/O? What is the main advantage of each?

Part III – CSE 320: Design and Synthesis of Digital Hardware

Problem 9

Register-Transfer Level (RTL) Design

A) What is the difference between a behavioral and a structural RTL description?

B) Describe the role of a finite state machine (FSM) in controlling a datapath, using a simple traffic light controller as an example.

Problem 10

Datapath and Control

A) List at least four key components of a typical datapath, and describe the role of the control unit in coordinating them.

Problem 11

Timing and Synthesis

A) Define setup time and hold time for a flip-flop, and explain why violating them can cause a circuit to malfunction.

B) What is the purpose of static timing analysis (STA) during hardware synthesis?

Part IV – CSE 420 Preview Concepts

Problem 12

A) C Program analysis

```
void foo(int *i){
      *i = *i + 1;
}

void main(void){
      int array [] = { 10, 20, 30, 40, 50 };
      int i, *ptr ;
      ptr = array;
      for ( i = 0 ; i <4 ; i++ ){
          foo(ptr++);
          printf ("\n%", *ptr ) ;
      }
}
```

What will be the output of the following program?

i) 11 21 31 41
ii) 20 30 40 50
iii) 21 31 41 51
iv) 10 20 30 40

B) C Program key semantics

```
#define MAX_NUM 15
```

Referring to the sample above, what is MAX_NUM?
i) MAX_NUM is a precompiler constant
ii) MAX_NUM is a preprocessor macro
iii) MAX_NUM is an integer variable
iv) MAX_NUM is a linker constant

Problem 13

The following describes a 5-stage pipelined processor. The pipelined processor should always compute the same results as an unpipelined processor. Answer the following questions for each of the instruction sequences below:

- Why does the sequence require special handling (what could go wrong)?
- What are the minimal hardware mechanisms required to ensure correct behavior?
- What additional hardware mechanisms, if any, could help preserve performance? Assume that the architecture does not have any branch delay slots, and assume that branch conditions are computed by the ALU.

A)

```
BEQ   r1, r0, 200    # branch to PC+200 if
r1 == r0
ADD   r2, r3, r5     # r2 <- r3 + r5
SUB   r4, r5, r6     # r4 <- r5 + r6
...
```

B)

```
ADD    r1, r0, r2     # r1 <- r0 + r2
SUB    r4, r1, r2     # r4 <- r1 - r2
...
```

C)

```
LD     r1, 0(r2)      # r1 <- Mem[r2+0]
ADD    r3, r1, r2     # r3 <- r1 + r2
...
```

Problem 14

A) With a 32-bit address, how many total Tag bits are required for a direct-mapped cache with 256 bytes of data and 16-byte blocks?
B) Give the block number that byte address 0x0018 maps to, for a 16-block direct-mapped cache with 16-byte blocks, where the Tag bits are the high-order bits.
C) Suppose we have a 2-way set-associative cache with 256 bytes of data, 16-byte blocks, and a 32-bit physical address, instead of a direct-mapped cache. Give the address breakdown for this cache in terms of Index, Tag, and Byte offset, and state how many bits each field occupies.

Problem 15

Describe the operation of a data cache. Your description should include discussion of the following:

A) Spatial and temporal locality.
B) Valid bits.
C) Direct-mapped versus set-associative structures. Show how cache indexing and tag match work for both direct-mapped and 2-way set-associative cache configurations, assuming one word per cache line. What are the advantages and disadvantages of direct-mapped versus set-associative structures?
D) Multiple-word cache lines. What are the advantages and disadvantages of multiple-word cache lines? Describe how they are implemented for a direct-mapped cache.
E) LRU and random replacement policies. What are their relative advantages and disadvantages?

Problem 16

The number 1.248 is equal to 1 + 2/10 + 4/100 + 8/1000. If we use a binary point instead of a decimal point, what will the following numbers equal in Base-10 and Base-16?

A) 0.101
B) 1.001
C) 00.101
D) 11.101
E) 1010.101

Problem 17

A) Describe the steps that transform a program written in a high-level language such as C into a representation that is directly executed by a computer processor. Name any intermediate representations that the code may take.
B) What is the difference between the stack and the heap?
C) What happens if the stack and heap segments collide (i.e., one overwrites part of the other)? Why is this dangerous? How would you prevent it?
D) What are the causes of a stack overflow? Give a specific example.
E) What is the difference between the .BSS and .DATA segments?

Problem Ratings

For each problem part, enter 0 (No idea), 1 (Used to know it), or 2 (Know it). Shaded cells indicate the problem does not have that part.

problem	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					

0 = No idea 1 = Used to know it 2 = Know it