CSC 137 Cokgor Homework 5 (5 points)
<u>Show your work. Work not shown will not earn points.</u>

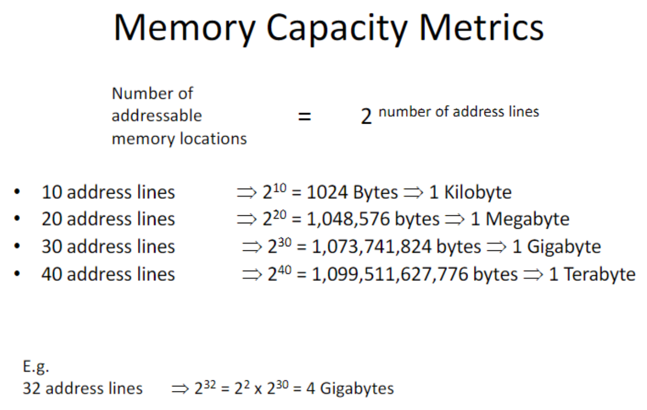
1) How many hexadecimal digits are needed to represent a binary number with 32 digits? (0.1 point)

ANSWER:
A single hexadecimal digit can represent a power of 16 (e.g.: 0,1,2,...,F). 4 binary digits can represent 16 numbers. We also get that $2^4 = 16$. So, we can divide 32 by 4 to get 8 hexadecimal digits! We get 8 hexadecimal digits from a 32-digit binary number!

2) A microprocessor can access any memory location by outputting the address of that location onto its address bus. You have designed a microprocessor and have made available 12 address lines for the address bus. You are using a memory device that is 8 bits wide (i.e. each memory location holds 1 Byte).

a. How many Kilobytes is your microprocessor's memory capacity? (State your answer in Kilobytes) (0.1 point)

ANSWER:
*Based on the slides, we know the following:



Memory Capacity Metrics

Number of addressable memory locations $= 2^{\text{number of address lines}}$

- 10 address lines $\Rightarrow 2^{10}$ = 1024 Bytes $\Rightarrow$ 1 Kilobyte
- 20 address lines $\Rightarrow 2^{20}$ = 1,048,576 bytes $\Rightarrow$ 1 Megabyte
- 30 address lines $\Rightarrow 2^{30}$ = 1,073,741,824 bytes $\Rightarrow$ 1 Gigabyte
- 40 address lines $\Rightarrow 2^{40}$ = 1,099,511,627,776 bytes $\Rightarrow$ 1 Terabyte

E.g.
32 address lines $\Rightarrow 2^{32} = 2^2 \times 2^{30}$ = 4 Gigabytes

12 address lines gives us (2^{10}= 1024) * (2^2= 4) = 4096 Bytes or 4 Kilobytes.
Our microprocessor's memory capacity is 4 kilobytes!

b. You have decided that you will need more memory capacity. How many address lines do you need to make available if you wanted to double the memory capacity? (0.1 point)

ANSWER:
Before:
12 address lines: (2^{10}= 1024) * (2^2= 4) = 4096 Bytes or 4 Kilobytes.
After:
13 address lines: (2^{10}= 1024) * (2^3= 8) = 8192 Bytes or 8 Kilobytes.

To double the memory capacity, we would just need to make one more address line available, or in this case make 13 address lines available!

3) How many address lines are required for a 2MByte memory? (0.1 point)

ANSWER:

20 address lines gives us (2^{20}= 1,048,576) = 1,048,576 Bytes or 1 Megabyte

To double the memory capacity, we just need to add one more address line.
So,

21 address lines: (2^{20}= 1,048,576) * (2^{1}= 2) = 2097152 Bytes or 2 Megabytes.

21 address lines are required for a memory of 2 Megabytes!

4) You will be building a memory with 14-bit address and 8-bit data word size. If this memory were constructed from 1Kb x 1-bit RAM chips, how many chips would be required? (0.2 point)

ANSWER:

If we have n-bit address and m-bit words then our RAM size will be 2^{n} x m.

14 address inputs = (2^{10}= 1024 bytes) * (2^{4}= 16) = 16384 bytes

RAM size = 16384 bytes x 1 byte (*8 bits) (*Required RAM size)

Number of chips required = Required RAM Size / Basic RAM Size

= 16384 x 1 / 1024 x 1

= 131072 / 1024

= 128 chips

5) Can you build a computer with non-volatile memory (e.g. ROM) only? Explain your answer. (0.1 point)

ANSWER:

You can build a computer with non-volatile memory only in that you can build embedded systems without RAM, like an automatic washing machine or a microwave oven. However, you cannot build a computer nor an embedded system without ROM. The reason for this is that ROM retains its contents even when the computer is turned off, whereas RAM would only hold its contents temporarily or lose it once power is turned off.

6) Can you build a computer with volatile memory (e.g. RAM) only? Explain your answer. (0.1 point)

ANSWER:

No, you cannot build a computer with volatile memory only. This is because you would need to be able to maintain system software, keep permanent system data, and such. You can build embedded systems without RAM, but you cannot build a computer nor an embedded system without ROM. So, the answer is no, you cannot build a computer with only volatile memory.

7) What does the Program Counter register (or the Instruction Pointer register) do? (0.1 point)

ANSWER:
The program counter register is a special-purpose register that contains the memory address of the next instruction to be executed. It is usually incremented after fetching some instruction, and would track the progress through the program by containing the address of the next instruction to be executed.

8) You are designing a simple microprocessor for a specific application. You have determined that you will need 32 unique operations in the instruction set. Each instruction needs two register fields to hold the operands. Each register field can address 16 internal registers.

a. How many bits does your op-code field need to be? (0.1 point)

ANSWER:
$Log_2(32)$ = 5 bits

b. How many bits does each of your register fields need to be? (0.1 point)

ANSWER:
$Log_2(16)$ = 4 bits

9) We have seen that fields of an instruction encoding can be mixed to achieve different goals depending on the instruction type. Refer to page 20 of the Instruction Set Architecture lecture notes. As examples, the Rm field of an R-format instruction is 5 bits, whereas the address offset field of a D-format instruction is 9-bits. Or, the opcode field of a D-format instruction is 11 bits, whereas the opcode field of an I-format instruction is 10-bits. Answer the following questions with field mixing in mind.
Assume the following instruction format:

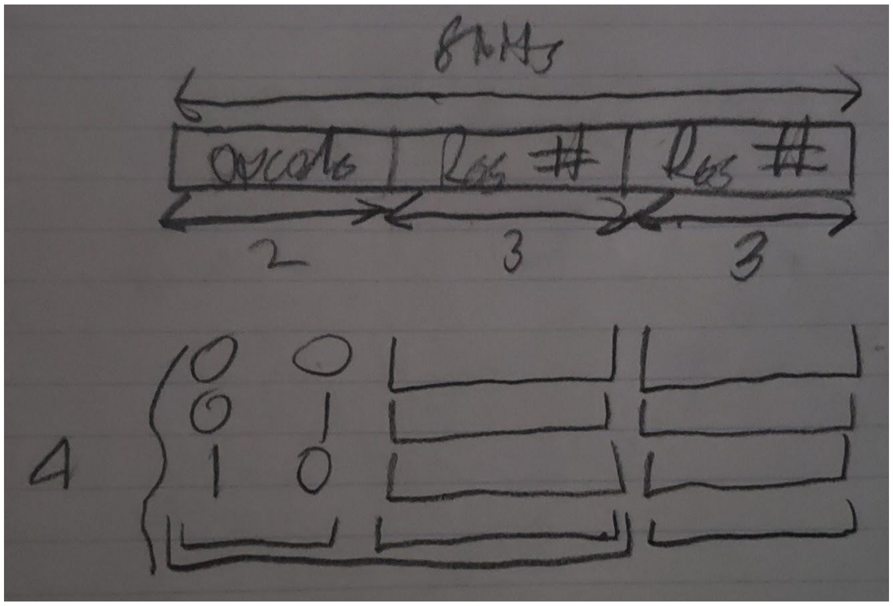
opcode	Rm	Rn

a. The overall instruction length is 8 bits, the register fields (R1, R2) are 3 bits each. How many unique opcodes can you have? (0.1 point)

ANSWER:
If the register fields are 3 bits each and there are 2 of them, then 6 bits are already used up out of the 8 total. This leaves us with 2 bits for the opcodes. We get 2^2 = 4 unique opcodes possible.

b. If you mix the fields, can you have 3 x two address and 8 one address instructions? Justify your answer. (0.2 point)

ANSWER:

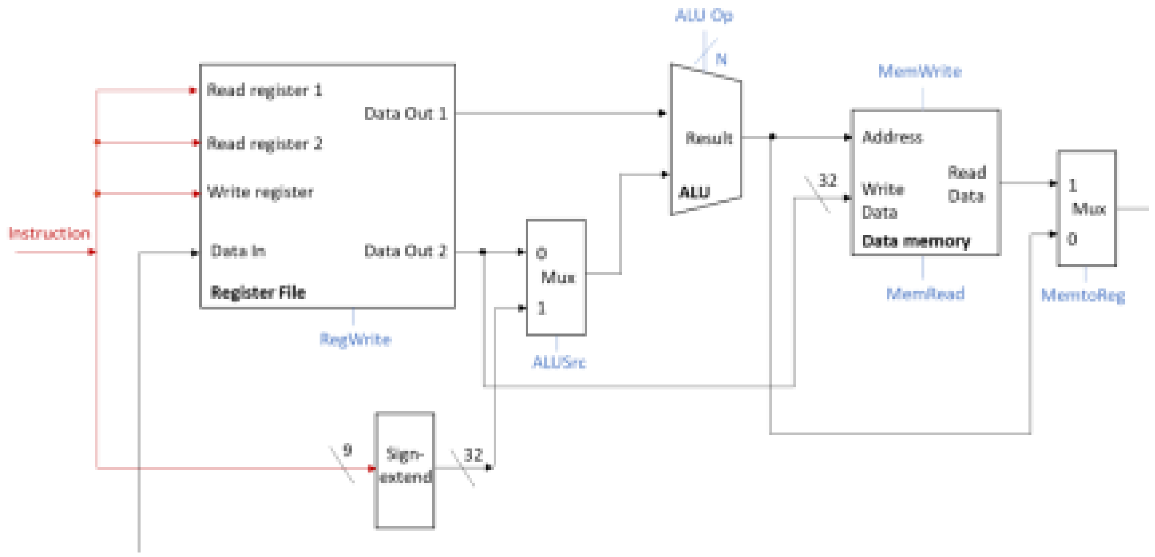


If you mix the fields, you can have 3 x two address instructions and you can have 8 one address instructions. So, if you look at the diagram above, you can see that from the perspective of the opcode, it is possible to have 3 x two address instructions with the opcode being the count and the registers being an address each. On the bottom row, if you look at it from the perspective of the rightmost register, then you can see that it is possible to have 8 one address instructions, with the rightmost register being the count and the rest being the address.

10) If the data storage is arranged according to the little-endian representation, how is the word 0x4CE3F90A stored in memory location 0x080001AC. (0.1 point)

Memory	Address
4C	0x080001AF
E3	0x080001AE
F9	0x080001AD
0A	0x080001AC
	0x080001AB
	0x080001AA
	0x080001A9

11) Consider the datapath developed during the class lectures for R-format and D-format instructions (shown below) and answer the following questions.



a. If the RegWrite control signal is stuck at 0 (i.e. not asserted), which instructions shown below would malfunction: (0.4 point)

i. ADD R0, R1, R2
ii. SUB R0, R1, R2
iii. LDR R0, [R1, #4]
iv. STR R1, [R0, #4]

ANSWER:
i,ii, and iii

b. If the MemWrite control signal is stuck at 0 (i.e. not asserted), which instructions shown below would malfunction: (0.4 point)

i. ADD R0, R1, R2
ii. SUB R0, R1, R2
iii. LDR R0, [R1, #4]
iv. STR R1, [R0, #4]

ANSWER:
iv

c. If the ALUSrc control signal is stuck at 0, which instructions shown below would malfunction: (0.4 point)

i. ADD R0, R1, R2
ii. SUB R0, R1, R2
iii. LDR R0, [R1, #4]
iv. STR R1, [R0, #4]

ANSWER:
iii, and iv

12) Consider the instruction execution timing table of a single cycle processor given below and answer the following questions:

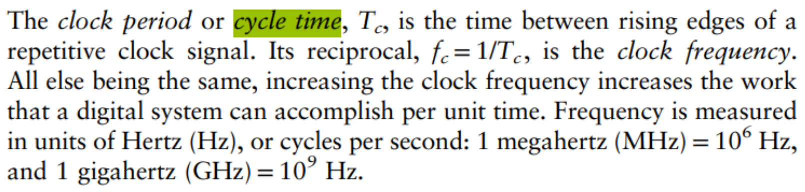
Instruction	Instruction Fetch	Register Read	ALU Operation	Data Access	Register Write	**Total Time**
LDR (Load Register)	200 ps	100 ps	200 ps	200 ps	100 ps	**800 ps**
STR (Store Register)	200 ps	100 ps	200 ps	200 ps		**700 ps**
ADD	200 ps	100 ps	200 ps		100 ps	**600 ps**
SUB	200 ps	100 ps	200 ps		100 ps	**600 ps**

a) If every instruction in the table takes exactly one clock cycle, what must be the clock cycle time to accommodate all instructions. (0.1 point)

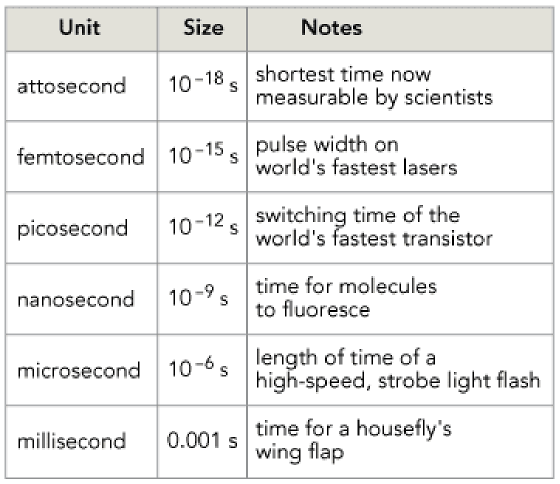
ANSWER:
800ps. This is because the clock cycle must be stretched to accommodate the slowest instruction, which would be LDR in this case (*especially if you compare the total times on the right of the table).

b) What would be the clock speed of the processor with the cycle time you calculated above? (0.1 point)

ANSWER:
*I used the excerpts below for reference:



The *clock period* or *cycle time*, T_c, is the time between rising edges of a repetitive clock signal. Its reciprocal, $f_c = 1/T_c$, is the *clock frequency*. All else being the same, increasing the clock frequency increases the work that a digital system can accomplish per unit time. Frequency is measured in units of Hertz (Hz), or cycles per second: 1 megahertz (MHz) = 10^6 Hz, and 1 gigahertz (GHz) = 10^9 Hz.



Unit	Size	Notes
attosecond	10^{-18} s	shortest time now measurable by scientists
femtosecond	10^{-15} s	pulse width on world's fastest lasers
picosecond	10^{-12} s	switching time of the world's fastest transistor
nanosecond	10^{-9} s	time for molecules to fluoresce
microsecond	10^{-6} s	length of time of a high-speed, strobe light flash
millisecond	0.001 s	time for a housefly's wing flap

800ps = 800 * 10^{-12}s = 8 x 10^{-10}
1/(8 x 10^{-10}) = 1,250,000,000 ≈1.3 GHz

The clock speed of the processor with the cycle time that I calculated above would be 1.3 GHz.

c) Assume that both instructions and data are in the same memory space, i.e. there is a single address and data bus pair for memory access. What will be the clock cycle time if you reduce the delay of the memory access by half. (0.1 point)

ANSWER:
If we reduce the delay of the memory access by half, then we should have an instruction execution timing table like the one below (*Changes from the previous table are highlighted in yellow).

Instruction	Instruction Fetch	Register Read	ALU Operation	Data Access	Register Write	**Total Time**
LDR (Load Register)	100 ps	100 ps	200 ps	100 ps	100 ps	**600 ps**
STR (Store Register)	100 ps	100 ps	200 ps	100 ps		**500 ps**
ADD	100 ps	100 ps	200 ps		100 ps	**500 ps**
SUB	100 ps	100 ps	200 ps		100 ps	**500 ps**

600ps. Again, this is because the clock cycle must be stretched to accommodate the slowest instruction, which is still LDR.

d) What would be the clock speed of the processor after the reduction of memory access time as in (c) above. (0.1 point)

ANSWER:
600ps = 600 * 10^{-12}s = 6 x 10^{-10}
1/(6 x 10^{-10}) = 1,666,666,667 ≈1.7 GHz

The clock speed of the processor with the cycle time that I calculated above would be 1.7 GHz.

13) Consider the instruction execution for a 5 stage pipelined processor for all instructions as shown below. Each phase takes 200 ps. Instruction Fetch and Data Access use the same memory. Assume that Reg Read and Reg Write phases do not conflict.

Instruction Fetch	Reg Read	ALU	Data Access	Reg Write

Answer the following questions:

a) Can the following program segment be executed without a pipeline hazard? If there is a hazard, then what type? (0.3 point)

```
LDR R1, [R0, #4]
ADD R6, R5, R1
```

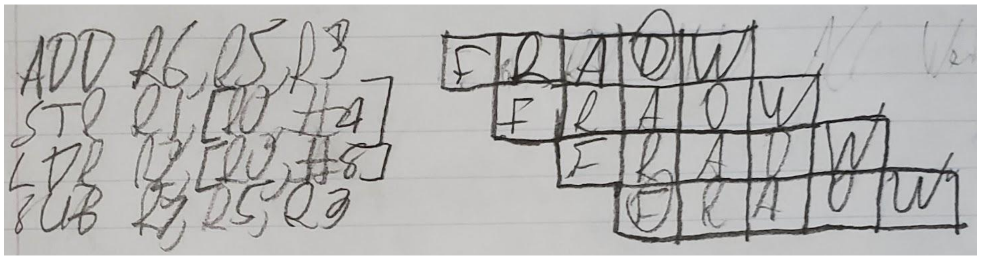
ANSWER:

The following program segment cannot be executed without a pipeline hazard. This program would give us a data hazard, which means that the pipeline must be stalled because one step must wait for another to complete. In this case, the add instruction is waiting on the load instruction to finish with one of its operands. The way I see it, load is loading something into R1. Because Add uses R1 as one of its operands, it would have to wait until load finishes or else there might be issues.

b) Can the following program segment be executed without a pipeline hazard? If there is a hazard, then what type? (0.3 point)

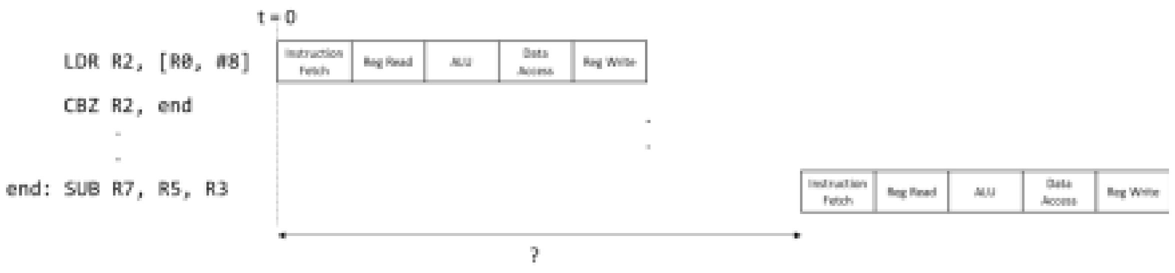
```
ADD R6, R5, R3
STR R1, [R0, #4]
LDR R2, [R0, #8]
SUB R7, R5, R3
```

ANSWER:

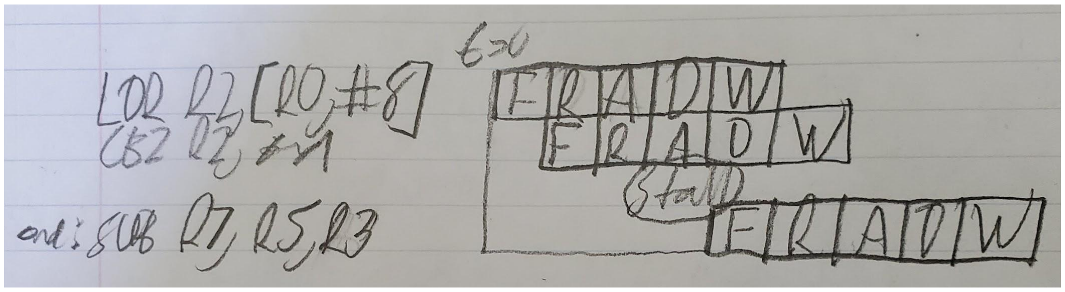


The following program segment cannot be executed without a pipeline hazard. This would give us a structural hazard. Assuming that Instruction Fetch and Data Access use the same memory, this would lead to a situation in which the combination of instructions cannot be executed in the same clock cycle. In this case, there would be a conflict between the Data Access for the add instruction and the Instruction Fetch for the subtract instruction.

c) The LDR instruction in the program segment shown below is fetched at t=0. If the branch is taken as a result of the CBZ instruction, when will the SUB instruction be fetched? Justify your answer by showing the pipeline. You must avoid any hazards by stalling the pipeline when necessary. (0.4 point)



ANSWER:



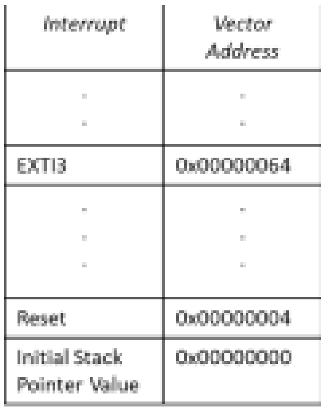
If the branch is taken as a result of the CBZ instruction, then the SUB instruction will be fetched once the load instruction starts its Reg Write phase and once the CBZ instruction starts its Data Access phase. You can also say that the SUB instruction will be fetched at 800ps.

14) Consider the memory map of a hypothetical processor below. Does this processor implement memory mapped I/O or isolated I/O? Explain your reasoning. (0.1 point)

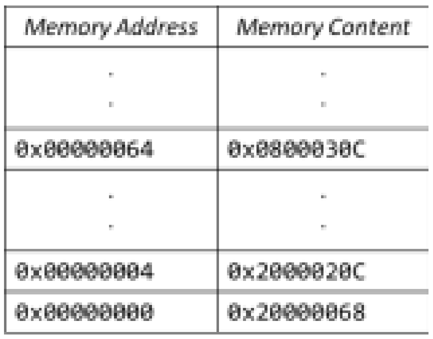
Flash
DRAM
Peripherals
SRAM
ROM

ANSWER:
This processor implements memory mapped I/O. The reason for this is because the memory map shown has its I/O device implemented as a memory location within it. Inside this memory map, there is a section for peripherals, which is the I/O portion of this memory map.

15) Given the Interrupt Vector Table and the memory contents below, answer the following questions: (0.4 point)



Interrupt	*Vector Address*
. .	. .
EXTI3	0x00000064
. . .	. . .
Reset	0x00000004
Initial Stack Pointer Value	0x00000000



Memory Address	*Memory Content*
. .	. .
0x00000064	0x0800030C
. .	. .
0x00000004	0x2000020C
0x00000000	0x20000068

a) Which address is the Interrupt Vector for Reset located?

ANSWER:
0x00000004

b) Which address is the Interrupt Vector for EXTI3 interrupt located?

ANSWER:
0x00000064

c) What is the interrupt vector for Reset?

ANSWER:
0x2000020C

d) What is the address of the EXTI3 Interrupt Service Routine?

ANSWER:
0x0800030C

16) Direct Memory Access (DMA):

A program needs to transfer a block of data from the computer hard drive to the main memory. A DMA request is initiated via the DMA controller. Answer the following questions based on what you learned in the class: (0.4 point)

a) Is this a DMA read or a DMA write transfer?

ANSWER:
This is a DMA write transfer because we are transferring data from a peripheral device to the memory.

b) Will the data that is being transferred be loaded to the microprocessor register, or will the data be transferred between the peripheral and the memory directly?

ANSWER:
The data that is being transferred will be transferred between the peripheral and the memory directly. If we are using a DMA transfer, then we wouldn't involve the microprocessor.

c) Is it the DMA controller or the processor that generates the corresponding hard drive and memory control signals?

ANSWER:
It is the DMA controller that generates the corresponding hard drive and memory control signals.

d) Can the processor initiate a separate transfer between an Ethernet port and the main memory during this data transfer?

ANSWER:
The processor can initiate a separate transfer between an Ethernet port and the main memory during this data transfer as this data transfer involves a technique that doesn't involve the microprocessor. DMA is also used for network communications.