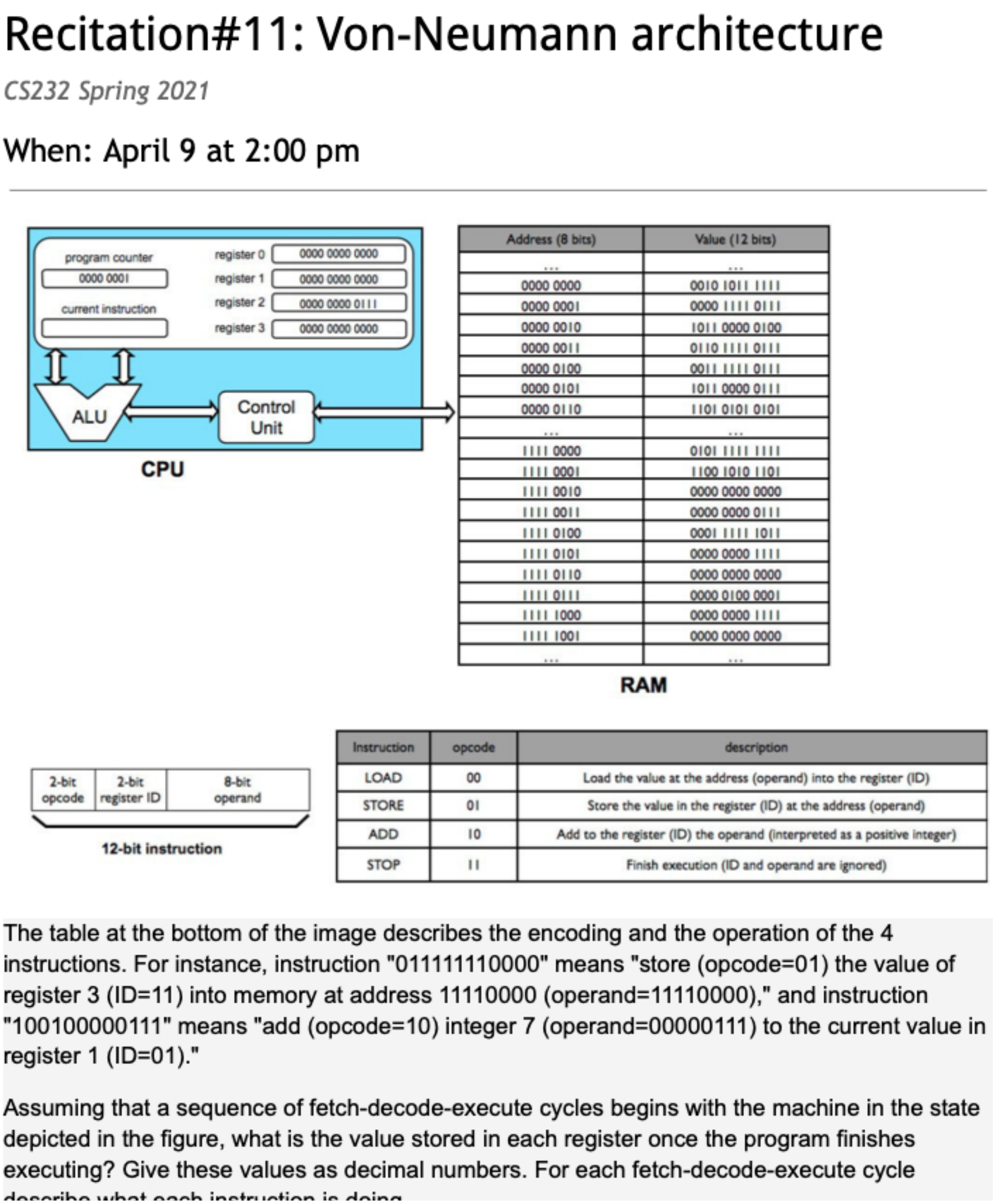
Week 11

Friday, April 9, 2021 13:09



Recitation#11: Von-Neumann architecture

CS232 Spring 2021

When: April 9 at 2:00 pm

Address (8 bits)	Value (12 bits)
...	...
0000 0000	0010 1011 1111
0000 0001	0000 1111 0111
0000 0010	1011 0000 0100
0000 0011	0110 1111 0111
0000 0100	0011 1111 0111
0000 0101	1011 0000 0111
0000 0110	1101 0101 0101
...	...
1111 0000	0101 1111 1111
1111 0001	1100 1010 1101
1111 0010	0000 0000 0000
1111 0011	0000 0000 0111
1111 0100	0001 1111 1011
1111 0101	0000 0000 1111
1111 0110	0000 0000 0000
1111 0111	0000 0100 0001
1111 1000	0000 0000 1111
1111 1001	0000 0000 0000
...	...

RAM

2-bit opcode	2-bit register ID	8-bit operand

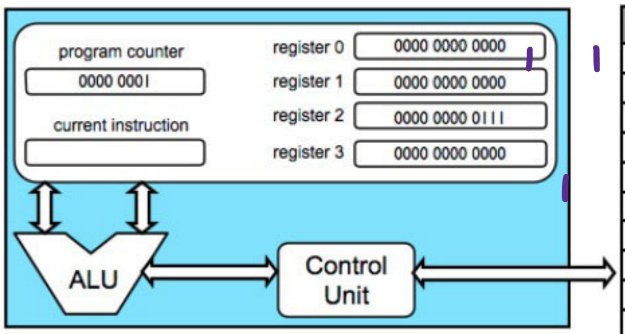
12-bit instruction

Instruction	opcode	description
LOAD	00	Load the value at the address (operand) into the register (ID)
STORE	01	Store the value in the register (ID) at the address (operand)
ADD	10	Add to the register (ID) the operand (interpreted as a positive integer)
STOP	11	Finish execution (ID and operand are ignored)

The table at the bottom of the image describes the encoding and the operation of the 4 instructions. For instance, instruction "011111110000" means "store (opcode=01) the value of register 3 (ID=11) into memory at address 11110000 (operand=11110000)," and instruction "100100000111" means "add (opcode=10) integer 7 (operand=00000111) to the current value in register 1 (ID=01)."

Assuming that a sequence of fetch-decode-execute cycles begins with the machine in the state depicted in the figure, what is the value stored in each register once the program finishes executing? Give these values as decimal numbers. For each fetch-decode-execute cycle describe what each instruction is doing.

describe what each instruction is doing.



CPU

Address (8 bits)	Value (12 bits)	
...	...	
0000 0000	0010 1011 1111	load value at _ into reg 10
0000 0001	0000 1111 0111	load val at _ into reg 0
0000 0010	1011 0000 0100	add _ to reg 11
0000 0011	0110 1111 0111	Store val in reg 10 to
0000 0100	0011 1111 0111	load val at _ to reg 11
0000 0101	1011 0000 0111	add _ to reg 11
0000 0110	1101 0101 0101	finish execution
...	...	
1111 0000	0101 1111 1111	
1111 0001	1100 1010 1101	
1111 0010	0000 0000 0000	
1111 0011	0000 0000 0111	
1111 0100	0001 1111 1011	
1111 0101	0000 0000 1111	
1111 0110	0000 0000 0000	
1111 0111	0000 0100 0001	
1111 1000	0000 0000 1111	
1111 1001	0000 0000 0000	
...	...	

RAM

2-bit opcode	2-bit register ID	8-bit operand

12-bit instruction

Instruction	opcode	description
LOAD	00	Load the value at the address (operand) into the register (ID)
STORE	01	Store the value in the register (ID) at the address (operand)
ADD	10	Add to the register (ID) the operand (interpreted as a positive integer)
STOP	11	Finish execution (ID and operand are ignored)