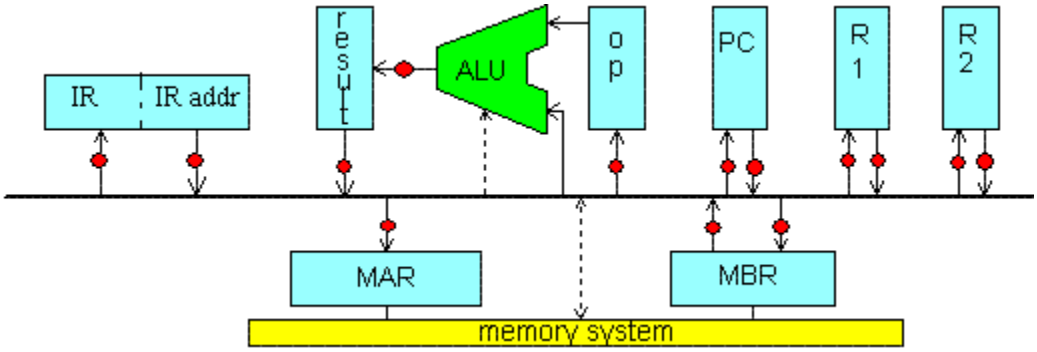
COMP375 Computer Architecture and Organization



The following table represents the microcode for the simple CPU shown above. Each column represents the possible setting of the switches shown as small circles in the diagram. The possible ALU functions are OPcode, ADD, SUB, INCrement and DECrement by 1. The possible memory functions are READ, WRITE, and WAIT.

1. **Memory Direct — Subtract R1, cow** Subtract the value in memory location cow from R1.

Instruction fetch and program counter increment shown in this problem, but not the following problems.

bus → IR	IRadr →bus	result →bus	ALU→ result	bus→ oprnd	bus→ PC	PC →bus	bus→ R1	R1 → bus	bus→ R2	R2 → bus	bus→ MAR	bus→ MBR	MBR →bus	ALU func	Mem func
To read the instruction for the instruction fetch, the program counter (address of the next instruction to be executed) is copied to the bus. This value is copied into the Memory Address Register and the memory system is instructed to read. Simultaneously, the ALU will increment the program counter value on the bus and save the result in the result register.															
			X			**X**					**X**			**inc**	**read**
The incremented program counter value is copied back to the program counter register. The CPU will not wait for the memory system to read the instruction which will be put in the Memory Buffer Register.															
		X			**X**										**wait**
The instruction that was red from memory is copied to the instruction register.															
X													**X**		
To read the memory operand (cow), the address is copied from the instruction address field to the memory address register. The memory system is told to read.															
	X										**X**				**read**
Getting ready for the subtraction, the value in R1 is copied to the operand register. The system now waits for the memory value to be put in the Memory Buffer Register by the memory system.															
				X				**X**							**wait**
The value from memory is copied from the Memory Buffer Register into the ALU and the two values are subtracted. The result is saved in the result register															
			X										**X**	**sub**	
The result is copied onto the bus and into R1.															
		X					**X**								

2. **Register Indirect** — add R1, [R2] Add to R1 the value in memory whose address is in R2.

bus → IR	IRadr →bus	result →bus	ALU→ result	bus→ oprnd	bus→ PC	PC →bus	bus→ R1	R1 → bus	bus→ R2	R2 → bus	bus→ MAR	bus→ MBR	MBR →bus	ALU func	Mem func
The address of the data operand is in register R2. This address is copied onto the bus and into the Memory Address Register to start the operand fetch. The memory is told to read.															
										X	X				read
An add requires two operand. The first value is copied from R1 into the operand register. Because the memory system is slower than the CPU, the CPU will have to wait for the memory to finish reading the other operand															
				X				X							wait
The memory operand is copied from the Memory Buffer Register into the ALU and added to the value in the operand register (from R1 by the previous line).															
			X										X	add	
The result is copied from the result register back to R1															
		X					X								

3. **jump someplace –** direct addressing

bus → IR	IRadr →bus	result →bus	ALU→ result	bus→ oprnd	bus→ PC	PC →bus	bus→ R1	R1 → bus	bus→ R2	R2 → bus	bus→ MAR	bus→ MBR	MBR →bus	ALU func	Mem func
The program counter always contains the address of the next instruction to be executed. Changing the value in the program counter changes what will be executed next. A direct memory jump just copies the address from the instruction into the program counter.															
	X				X										