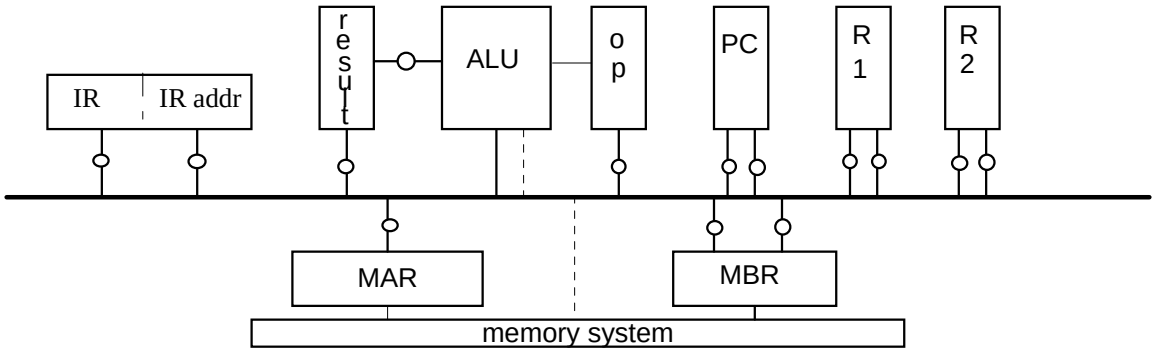
CS375 Computer Architecture and Organization



The microcode for the simple CPU shown above can be represented as a table. Each column represents the possible setting of the switches shown as small circles in the diagram. The possible ALU functions are ADD, SUB, AND, OR, and increment by 1. The possible memory functions are READ, WRITE, and WAIT.

The following series of micro instructions implements an ADD instruction: **ADD R1,xyz**
This instruction adds the contents of register 1 with the contents of the memory location xyz and stores the result back in register 1. We will assume direct addressing (the complete memory address is specified in the address field of the instruction).

bus → IR	IRadr →bus	result →bus	ALU→ result	ALU funct	bus→ oprnd	bus→ PC	PC →bus	bus→ R1	R1 → bus	bus→ R2	R2 → bus	bus→ MAR	bus→ MBR	MBR →bus	Mem funct
			X	inc			X					X			read
		X				X									wait
X														X	
	X											X			read
					X				X						wait
			X	add										X	
		X						X							

The following series of micro instructions implements a JUMP or BRANCH instruction:

JUMP nextinstr

where nextinstr is the address of the next instruction to be executed. We will assume direct addressing.

bus → IR	IRadr →bus	result →bus	ALU→ result	ALU funct	bus→ oprnd	bus→ PC	PC →bus	bus→ R1	R1 → bus	bus→ R2	R2 → bus	bus→ MAR	bus→ MBR	MBR →bus	Mem funct
			X	inc			X					X			read
		X				X									wait
X														X	
	X					X									

The following series of micro-instructions implements an **ADD** instruction. We will use memory indirect addressing (the operand field of the instruction contains the address of a memory location that contains the address of the data).

bus → IR	IRadr →bus	result →bus	ALU→ result	ALU funct	bus→ oprnd	bus→ PC	PC →bus	bus→ R1	R1 → bus	bus→ R2	R2 → bus	bus→ MAR	bus→ MBR	MBR →bus	Mem funct
			X	inc			X					X			read
		X				X									wait
X														X	
	X											X			read
															wait
												X		X	read
					X				X						wait
			X	add									X		
		X						X							