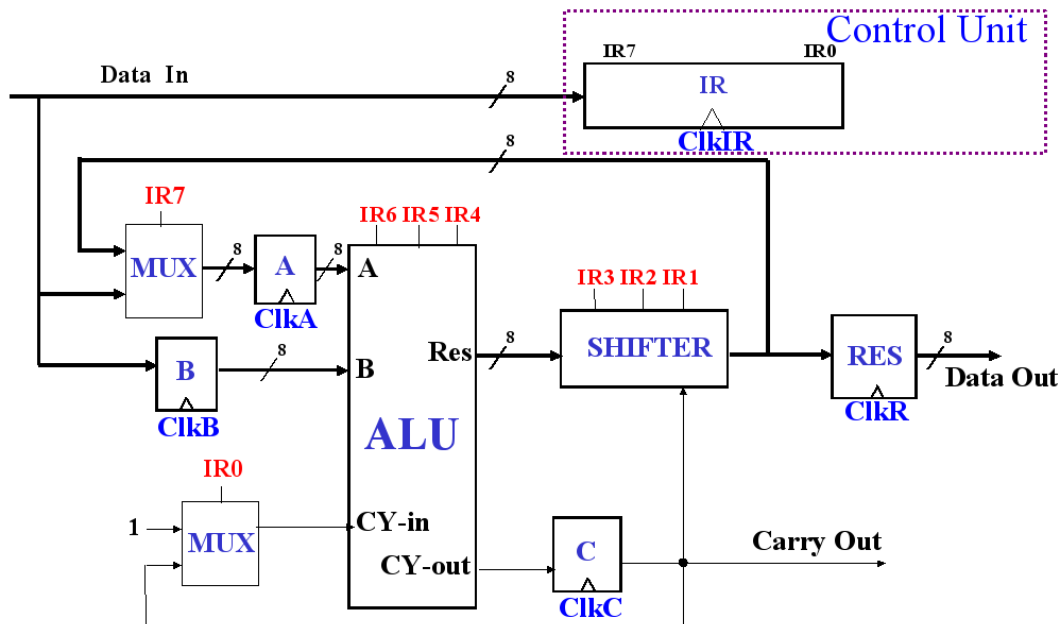


Tutorial 9: The Manual Processor

In lectures 14 and 15 we developed a simple eight-bit externally programmed computer. We now examine in detail how it works.



Selection Bits	000	001	010	011	100	101	110	111
ALU Function	0	B-A	A-B	A plus B	A xor B	A+B	A.B	-1
Shifter Function	Unchanged	Shift left Cin=0	Rotate Right	Shift Right Cin=0	Unchanged	Shift Left Carry	Shift Right Cin=Data[7]	Shift Right Carry

The basis of all digital computer operations is the register transfer operation. When the clock is applied one or more register transfer operations take place. The type and number of register transfer operation(s) are controlled by the bits stored in the Instruction Register or IR. The bits in this register perform the following functions:

1. Determine the functionality of controlled function generators such as the ALU and Shifter.
2. Set the select lines of multiplexers to connect the required registers . For example, if IR7 is zero the shifter is connected to register A. If it is one the data input bus is connected to A.

In addition, registers are updated by the application of clock signals. During the proper operation of the processor, the following clock signals are generated:

State	1	2	3	4	5
Clock	IR	A	B and C	IR	RES and C

This sequence is repeated as long as the processor is running. Now we have enough information to specify the bits in the input data which are loaded into the IR during clock times 1 and 4. For example:

ADD: Add two numbers together

State	Register transfer operations	Multiplexer and Function Selection
1	IR ← DataIn (Opcode 1)	Does not matter
2	A ← DataIn	A connected to DataIn
3	B ← DataIn and C ← 0	ALU=0 and CYout=0
4	IR ← DataIn (Opcode 2)	Does not matter
5	Res ← A pl B and C ← ALUCarry	ALU=plus, Shifter=unchanged, CY-in connected to C

Thus, we can specify the input data necessary to do the required instruction. Notice that the transfer takes place at the end of the state; therefore, the contents of the IR change just before states 2 and 3, as shown below.

		Instruction register (IR)							
State	Data In	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
1	1000XXXX	X	X	X	X	X	X	X	X
2	Number 1	1	0	0	0	X	X	X	X
3	Number 2	1	0	0	0	X	X	X	X
4	X0110001	1	0	0	0	X	X	X	X
5	XXXXXXXXX	X	0	1	1	0	0	0	1

The select line of the multiplexer which controls the ALU/CY-in signal (IR0) is a "don't care" for steps 1 to 4 because the ALU function selection is 000 (IR6-IR4) and the ALU provides output 00000000 disregarding what the carry input to the ALU is. As shown on the diagram, this bit if set to 0 selects the logic 1 input, and if set to 1 selects the the C input.

In the problems below you are asked to specify the required data inputs for the five states in order to execute some computer operations. Indicate as many "don't care" bits as possible since this may reduce the complexity of the outside circuit which controls this externally controlled processor.

Problem 1: INC(Increment) Result = Number + 1

		Instruction register (IR)							
State	Data In	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
1									
2									
3									
4									
5									

Problem 2. FN1(A+B-C)/2 Result = (Number1+Number2-Number3)/2

(This will require two complete five-state cycles, and you may assume that no carry is generated)

		Instruction register (IR)							
State	Data In	IR7	IR6	IR5	IR4	IR3	IR2	IR1	IR0
1									
2									
3									
4									
5									
1									
2									
3									
4									
5									

Problem 3

Many complex instructions would require the loading of the B register from the A register. Suggest ways in which the processor hardware could be re-designed such that this becomes possible.