SIES Graduate School of Technology

Subject :

Roll No.:

Assignment No. 2

Date : / /

i) State functions of Control unit of processor & explain Micro-Programmable CU

A a) Function of CU

i) It coordinates the sequence of data movement into, out of & between a processor's many sub-units.

ii) It interprets instruction.

iii) It receive external instructions or commands to which it converts to sequence of control signal

iv) It controls data flow inside the processor

v) It controls many execution units (i.e. ALU, data buffers & registers) contained within a CPU

vi) It also handles multiple tasks, such as fetching, decoding, execution, handling & storing results

b) Microprogrammable Control Unit :-

i) In microprogrammed control units, subsequent instruction words are fetched into the instruction register in a normal way. However, the operation code of each instruction is not directly decoded to enable immediate control signal generation but it comprises the initial address of a microprogram contained in the control store.

ii) The fundamental difference between these unit structures and the structure of hardwired CU is the existence of the CU that is used for storing words containing encoded control signals mandatory for instruction execution.

2) i) The 'next' micro instruction is loaded into the MIR and the MIR sends out control signal. The control signals that is important in phase 1 are A & B fields, which select the register R1(0001) & R2(0010)

ii) A- & B- latches are updated & captures the values of R_1 8 R_2 8 keep these values for the remainder of CPU cycle

iii) Now, ~~the~~ the ALU 8 Shifter gets times ~~out~~ to do the computation, in this example, the ALU will output the first input (= R_1) 8 the Shifter will not shift, so that the ~~result~~ result R1 is produced on the C-bus 8 simultaneously, MAR is updated with the value of B-latch because MAR bit in the micro-instructions is 1 8 allows phase 3 clock trigger update to the MAR

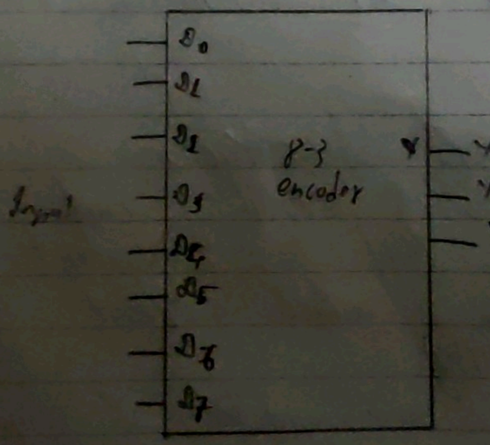
iv) ~~It~~ ~~8~~ Now, the result on C-bus (=R1) is written to destination. The destination ~~in~~ in example is MBR 8 not any register

v) In summary, R1 is copied to MBR 8 R2 into MAR

3) Octal to binary

i) An octal to binary encoder consist of 8 inputs 8 3 output lines. Each input line corresponds to each octal digit 8 3 output generate corresponding binary code

ii) In encoder, it is ~~about~~ about to be assumed that only one input is active or has a value 1 at given time otherwise, the circuit has no meaning. ~~8~~



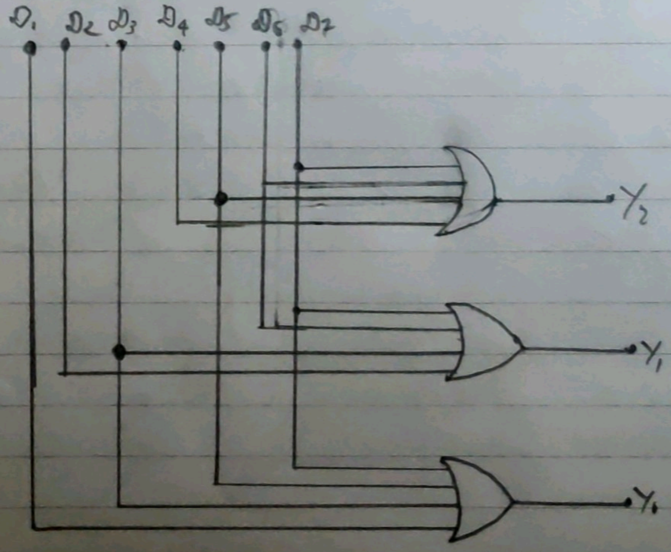
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

Now,

$Y_2 = D_4 + D_5 + D_6 + D_7$

$Y_1 = D2 + D_3 + D_6 + D_7$

$Y_0 = D_1 + D_3 + D_5 + D_7$



4)i) When J, K & clock are equal to 1, toggling takes place. Hence, propagation delay has also been reduced so output will be given out @ istance input is given. So there is a toggling again. Therefore, whenever clock=1, there are consecutive toggling. This is called Race around

Methods

a) Increasing delay of flip-flop

b) Use of edge triggered flip-flop

c) Using master-slave flipflop.