DIGITAL LOGIC AND PROCESSOR

Experiment 9 **Microprogram Controller** **Monsoon 2011**

The execution of every instruction in a program, written and stored by the user in the Program Memory, is achieved by performing a sequence of steps, each step requiring specific hardware operations like data transfer, ALU operation, program flow control etc. The set of values of the control bits needed at any step by the various hardware blocks of the Processor defines a Microinstruction. The task of generating the correct sequence of Microinstructions constituting the Microprogram required for the execution of an Instruction is handled by the **Microprogram Controller**, consisting of the **Instruction Decoder,** the **Microsequencer** and the **Microprogram Memory**.

The process of execution of an instruction is initiated by the Fetch cycle, which consists of the following three steps, taking place at successive active edges of the Clock pulse.

(i) Send the Instruction Address from PC to the Memory Address Register (MR): [MR] ← [PC].

(ii) Read the Op(erational) Code of the Instruction from the addressed Memory location and load it into the Instruction Register (IR): [IR] ← [[MR]].

(iii) Decode the Op Code (OC) by the Instruction Decoder to generate the Starting Address (SA) of the Microprogram for the Instruction to be executed, and load SA into the Microsequencer (MS), which provides the Microprogram Memory Address (MA) at its output: MA ← SA.

The output of the Microprogram Memory is connected to all the hardware blocks according to the control bits needed by them. Thus the Microinstruction given out by the Microprogram Memory in response to the address MA provides all the control action required at each step. Note that the Microprogram Memory, being a Read-Only Memory (ROM), does not require any Clock pulse and is thus combinational in nature. One can therefore design the Microprogram Memory using gates, as we will actually be doing in this experiment.

The Fetch cycle presupposes that the Address of the location where the Instruction has been stored in the Program Memory is available in the Program Counter (PC), and that the Microsequencer (MS) has been initialised to give MA = 000….00. The Fetch cycle is followed by the Execution cycle, which starts with the Microsequencer (MS) loaded to give MA = SA, and consists of as many steps as necessary for performing the task specified by the Instruction. As in the Fetch cycle, the Microinstruction given out by the Microprogram Memory in response to the address MA generated by the Microsequencer provides all the control action required at each step.

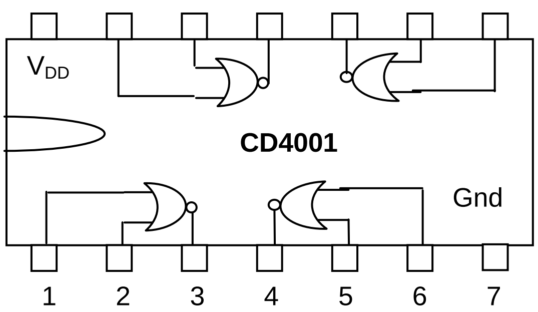
The values of the control bits R_{MS} and L_{MS} in each Microinstruction decide the flow of the Microprogram as follows:

If R_{MS} = 1, MA ← 000….00 at the next active edge of the Clock.

If L_{MS} = 1, MA ← SA at the next active edge of the Clock.

In this experiment, we will design an elementary Microprogram Controller, capable of handling just four instructions I_0, I_1, I_2, I_3. The Instruction Decoder will thus receive a 2-bit Op Code OC_{1-0} and generate a 4-bit Microprogram Starting Address SA_{3-0}, which will be used by the Microsequencer to generate a 4-bit Microprogram Memory Address MA_{3-0} as shown in Fig. 9.1. Table 9.1 gives the Op Code assignments and the Microprogram Address assignments for the four instructions. The Fetch cycle, of course, must have the Microprogram Address assignment as shown in the table. As no other hardware block of the Processor like registers or ALU will be used, each Microinstruction will only consist of two bits – R_{MS} and L_{MS}. The main objective of this experiment will be to study how a Microprogram Controller controls the flow of microprograms by the control bits R_{MS} and L_{MS}.

CD4011(NAND) and CD4081 (AND) are similar



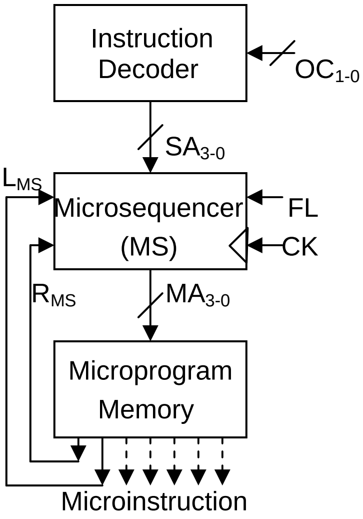


Fig. 9.1 Microprogram Controller

Table 9.1 Microprogram Memory Map

Instruction /Fetch Cycle T	Op Code OC$_{1-0}$	Microins-tructions	Start Addr. SA$_{3-0}$	End Address
a Fetch	-	3	0000	0010
b I_0	00	4	0011	0110
l I_1	01	3	0111	1001
e I_2	10	2	1010	1011
I_3	11	4	1100	1111

1. The Microsequencer will have to be realised with the presettable counter CD4029, used earlier in Experiment 6. As this counter has a provision for presetting the count Q_{3-0} to any value through the input D_{3-0}, but no separate provision to clear the count to 0000, we will have to realise the functions of the control bits R_{MS} and L_{MS} by incorporating the following logic (taking FL = 0 always):

$$D_{3-0} = SA_{3-0} \text{ if } L_{MS} = 1, \text{ and } D_{3-0} = 0000 \text{ if } R_{MS} = 1;\ SEN = L_{MS} + R_{MS}.$$

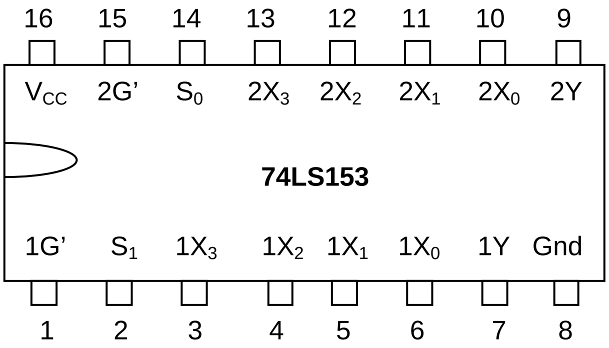
Connect the counter control inputs CI', U/D' and B/D' to V_{CC} / Gnd appropriately to obtain a binary up counter. Design the necessary combinational logic circuits using 2-input NOR/NAND/AND gates.

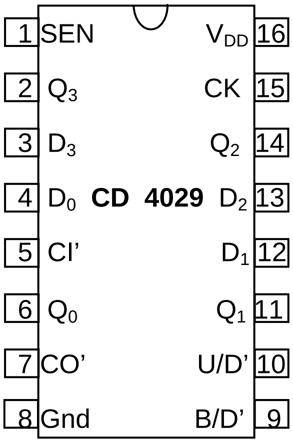
2. Design the Instruction Decoder so that it satisfies the relationship between OC and SA given in Table 9.1. Obtain Boolean expressions for the four bits comprising SA_{3-0} in terms of the Op Code bits OC_{1-0} and realise them using 2-input NOR/NAND/AND gates.

3. Given that R_{MS} = 1 only in the last Microinstruction of each Microprogram, and L_{MS} = 1 only in the last Microinstruction of the Fetch cycle, construct the K-map for the two Microinstruction bits R_{MS} and L_{MS} in terms of MA_{3-0}, and generate R_{MS} and L_{MS} using the two 4-input multiplexers available in the 74LS153 chip, used in Experiment 4. Use MA_3 and MA_2 as the Select inputs of the multiplexers and generate the multiplexer inputs from MA_1 and MA_0 with 2-input NOR/NAND/AND gates.

4. Decide on the final implementation of all the combinational circuits so that the complete circuit can be made using one each of the CD4001, CD4011 and CD4081 chips.

5. Assemble the complete circuit and connect MA_{3-0}, R_{MS} and L_{MS} to LED Displays. Connect the Op Code input OC_{1-0} to two Input Switches. Apply Manual Clock pulses to the Microsequencer Clock input and tabulate the sequence of MA_{3-0}, R_{MS} and L_{MS} generated by the Microprogram Controller for each value of OC_{1-0}, applied from the two input Switches. Verify that the observed sequence is according to Table 9.1.


16
15
14
13
12
11
10
9
V$_{CC}$
2G'
S$_0$
2X$_3$
2X$_2$
2X$_1$
2X$_0$
2Y
74LS153
1G'
S$_1$
1X$_3$
1X$_2$
1X$_1$
1X$_0$
1Y
Gnd
1
2
3
4
5
6
7
8


1 SEN
V$_{DD}$ 16
2 Q$_3$
CK 15
3 D$_3$
Q$_2$ 14
4 D$_0$
CD 4029
D$_2$ 13
5 CI'
D$_1$ 12
6 Q$_0$
Q$_1$ 11
7 CO'
U/D' 10
8 Gnd
B/D' 9