Experiment 6 Bus-based ALU-Register Data Transaction Monsoon 2018

In this experiment, we will set up and study the simple bus-based digital system shown in Fig. 8.1, consisting of a 74LS181 ALU, two Registers (CD4035 and CD4076), and a Tri-State Buffer (74LS125).

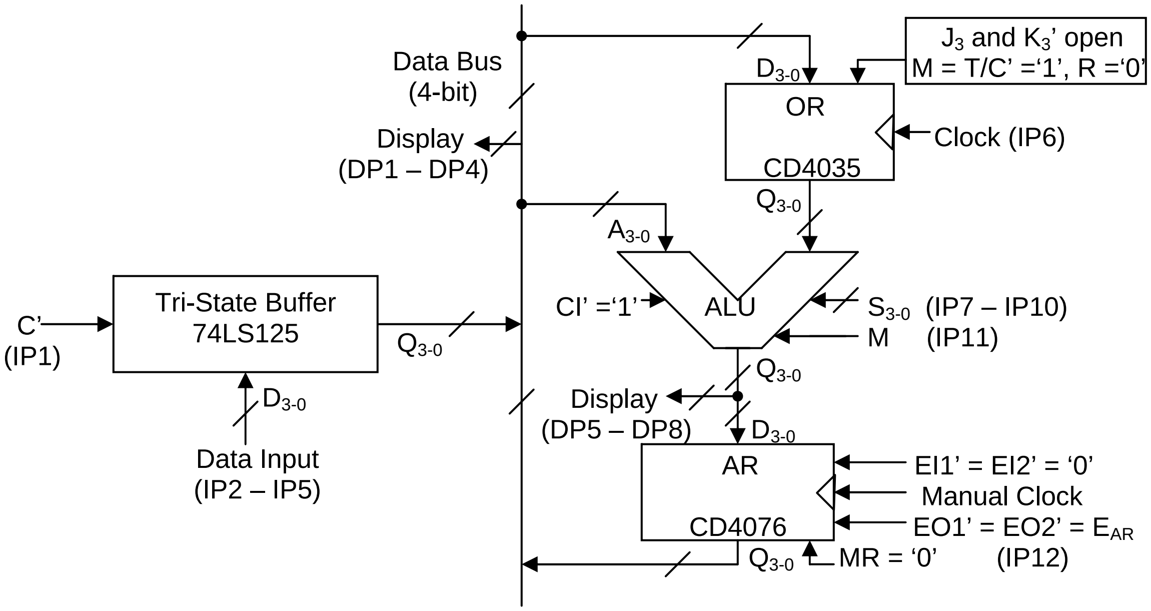
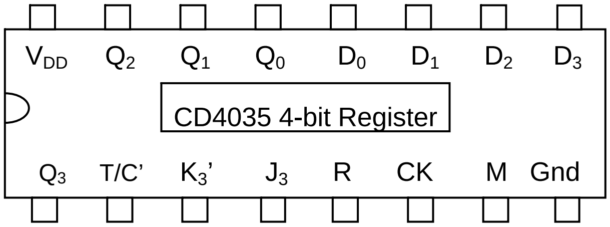
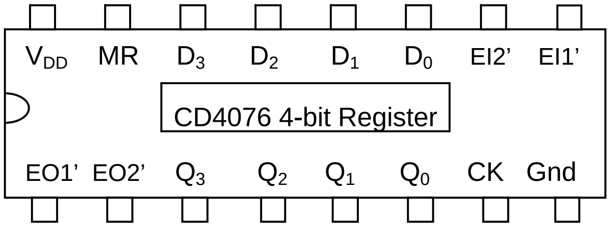


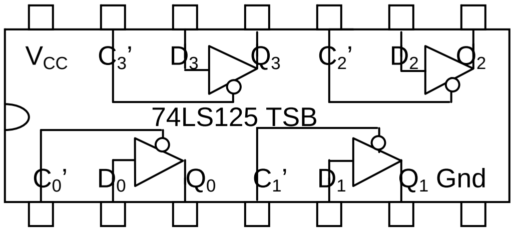
Fig. 8.1 Programmable ALU with Registers on a Bus

The pin connections of the CD4035, CD4076, 74LS125 and 74LS181 chips are given below. The CD4035 serial/parallel shift register will be used (with M = 1) as the Operand Register (OR) in this experiment. The CD4076, a 4-bit register with separate 2-bit controls for input and output, will be used as the Accumulator (AR) to store the ALU output after every ALU operation. The 74LS125, a quad Tri-State Buffer (TSB), will be used as a 4-bit buffer with a common control. The 74LS181 is a programmable ALU having 16 logic functions and 16 arithmetic functions as listed in Table 8.1.

For all the chips, Q_{3-0} denotes the 4-bit data output. For the registers (OR and AR) and the TSB, D_{3-0} denotes the 4-bit data input, while A_{3-0} and B_{3-0} denote the two 4-bit data inputs applied to the ALU. The CD4076 enables its 4-bit input D_{3-0} to be loaded into the four flip-flops at the next rising edge of the clock (CK) input if EI1'+ EI2' = 0, and enables its four flip-flop outputs to appear at the 4-bit output Q_{3-0} if EO1'+ EO2' = 0, the Q_{3-0} bits remaining "floating" (high-impedance) if EO1'+ EO2' = 1. MR (Master cleaR) is an asynchronous clear input for all the flip-flops in the register. EI1', EI2' and MR will be kept grounded throughout this experiment so as to permit the ALU output to be stored in AR at every Manual Clock. Each buffer (i = 0,1,2,3) of the 74LS125 has a negative-logic control input C_i': $Q_i = D_i$ if C_i' = 0 and Q_i remains "floating" (high-impedance) if C_i' = 1. In this experiment, the 74LS125 will be used as a 4-bit TSB by making the control common: C' = C_0' = C_1' = C_2' = C_3', and applying it from an Input Switch (IP1).







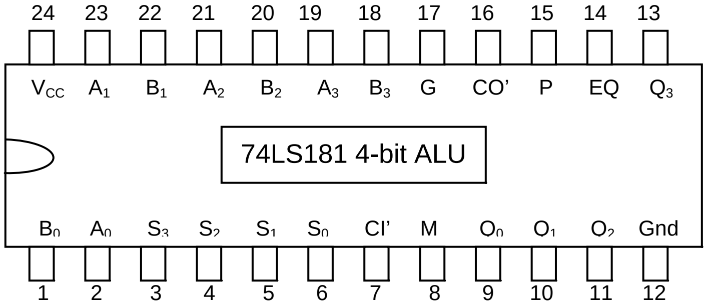


Table 8.1 Function Table of the Programmable ALU 74LS181

S_{3-0}	M=HIGH (Logic)	M=LOW (Arithmetic)	S_{3-0}	M=HIGH (Logic)	M=LOW (Arithmetic)
0	A’	A PLUS CI	8	A’+B	A PLUS (A•B) PLUS CI
1	(A+B)’	(A+B) PLUS CI	9	(A⊕B)’	A PLUS B PLUS CI
2	A’•B	(A+B’) PLUS CI	10	B	(A+B’) PLUS (A•B) PLUS CI
3	0000	Zero MINUS CI’	11	A•B	(A•B) MINUS CI’
4	(A•B)’	A PLUS (A•B’) PLUS CI	12	1111	A PLUS A PLUS CI
5	B’	(A+B) PLUS (A•B’) PLUS CI	13	A+B’	(A+B) PLUS A PLUS CI
6	A⊕B	A MINUS B MINUS CI’	14	A+B	(A+B’) PLUS A PLUS CYI
7	A•B’	(A•B’) MINUS CI’	15	A	A MINUS CI’

Note the following conventions to be followed while assembling the circuit in a step by step manner:

(i) The four horizontal bus strips available on the breadboard will be used as the four lines of the Data Bus, in the order 3-2-1-0 from top to bottom.
(ii) All connections to the Bus and to the LED Displays DP1-DP8 will be made with ***blue*** wires.
(iii) Input Switch connections to the TSB and the ALU will be made with ***green*** wires.
(iv) All other connections will be made with ***yellow*** wires.
(v) The exact allocations for the Input Switches and LED Displays have been given in Fig. 8.1.

1. Assemble the TSB circuit on the left side of the vertical Data Bus shown in Fig. 8.1. Make C’ = 1 and note that all the LEDs in DP1-DP4 are OFF, indicating that the bus is “floating”, i. e. not “driven” to a proper logic level by a digital output. Enable the TSB output by making C’ = 0 and apply different data inputs through IP2-IP5 to verify that DP1-DP4 correctly display the applied input.

2. Complete the interconnections to the OR and the ALU next, ***leaving out the connections to the AR***. Transfer a suitable 4-bit number from IP2-IP5 to OR by enabling TSB output (C’ = 0) and applying a Clock pulse to OR through IP6. Note that contact bounce is not a problem here, as multiple clock pulses would simply load the same input to the register again and again. Apply another 4-bit number from IP2-IP5 to the Bus and select an ALU function according to Table 8.1 by applying the appropriate ALU control inputs S_{3-0} and M through IP7-IP11, keeping C’ LOW. Verify that the 4-bit number displayed on DP5-DP8 does give the correct ALU output corresponding to inputs A_{3-0} coming from the Bus and B_{3-0} coming from OR. Tabulate the inputs applied and the output observed for at least four logic and four arithmetic functions.

3. Complete the interconnections to the AR. The circuit is now ready for sequential data transactions through the bus, but one has to be extremely cautious in this part of the experiment to avoid ***BUS CONFLICT***, which can happen if the two possible sources for data to be put on the bus – the TSB and AR – are both enabled simultaneously, i.e. if C’ and E_{AR}’ are both LOW at the same time. To avoid accidental mistakes, you should switch OFF power every time you make any change in the connections or Input Switch settings, and ***switch power ON only after ensuring that C’ (IP1) and E_{AR}’ (IP12) are not both LOW.***

4. Transfer a suitable 4-bit number from IP2-IP5 to OR and apply another 4-bit number from IP2-IP5 to the Bus as done in step 2, ***keeping C’ LOW and E_{AR}’ HIGH***. Choose a suitable ALU function and note the ALU output displayed on DP5-DP8, as done in step 2. Apply a Manual Clock pulse to AR, causing the ALU output to be loaded in AR. Now disable the TSB output and enable the AR output by ***making C’ HIGH first and then E_{AR}’ LOW***. Note that the ALU output changes immediately, because the input A_{3-0} is now coming from the AR output. Verify that DP1-DP4 now displays the same number as displayed earlier on DP5-DP8.

5. Perform a sequence of 3-4 suitably chosen ALU operations with the same number in OR and observe the complete sequence of ALU outputs shown by DP5-DP8 and AR outputs shown by DP1-DP4. Tabulate the results and verify the values displayed at each step theoretically.

6. Plan and demonstrate steps to add a sequence of numbers applied through IP2-IP5.