Datapath

By Sidhardh Burre

CS/ECE 2330 Digital Logic Design

April 19, 2021

Table of Contents

Contents

Table of Figures

Table of Tables

1 Problem Statement

Design the datapath as shown in Figure 1, which will complete the Central Processing Unit (CPU) design.

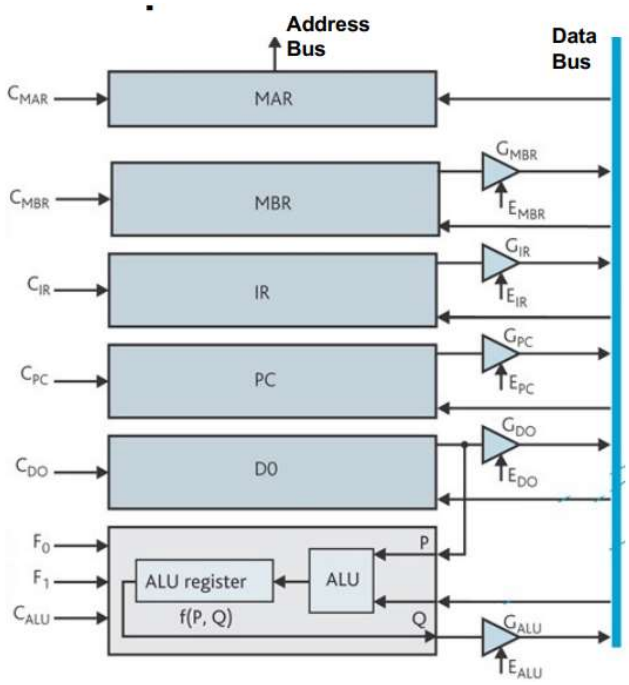

Figure 1: Simple CPU Datapath

The function of the datapath is to store all the information that is used to define the state of the CPU at any given time. This information is stored in the Memory Address Register (MAR), the Memory Buffer Register (MBR), the Instruction Register (IR), the Program Counter (PC), and Data register 0 (D0). In addition, the datapath contains an Arithmetic and Logic Unit (ALU) that is used to perform the 4 arithmetic operations shown in Table 1. Finally, note that there is an additional ALU register used to store the result of the arithmetic operation defined by F1 and F0.

F_1	F_0	Operation
0	0	Add P to Q, P + Q
0	1	Subtract Q from P, P – Q
1	0	Increment Q, Q + 1
1	1	Decrement Q, Q – 1

Table 1: Arithmetic Functions Defined by F0 and F1

2 Analytical Design

Employing the datapath diagram displayed in Figure 1, a datapath was designed that allowed for the proper registers to be activated. The resulting circuit schematic is shown below in Figures 2 and 3.

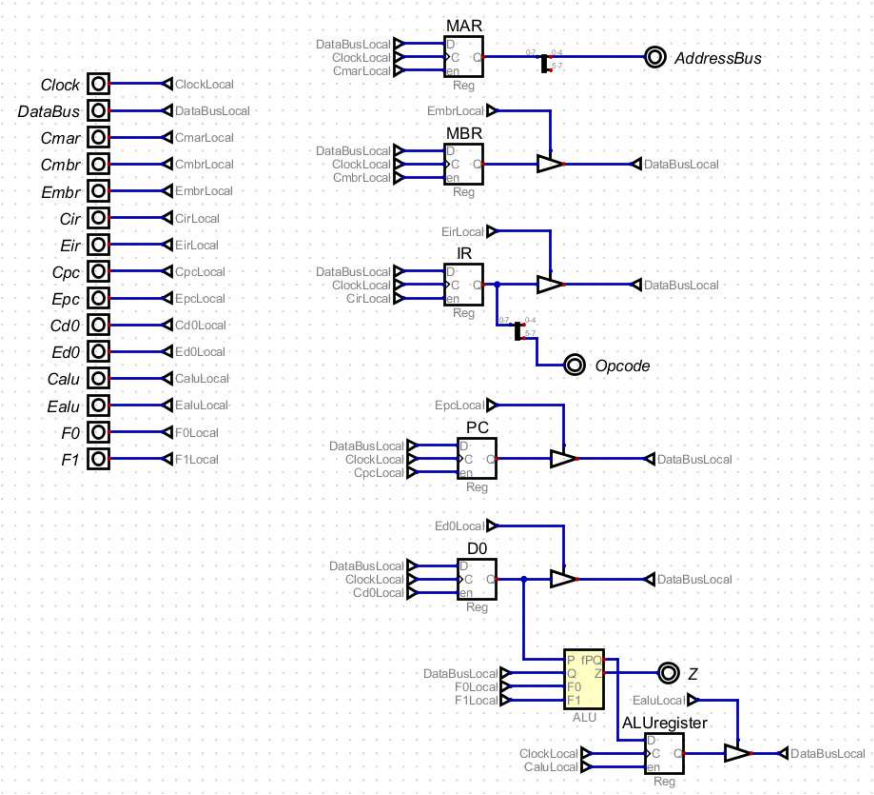

Figure 2: Circuits for Datapath

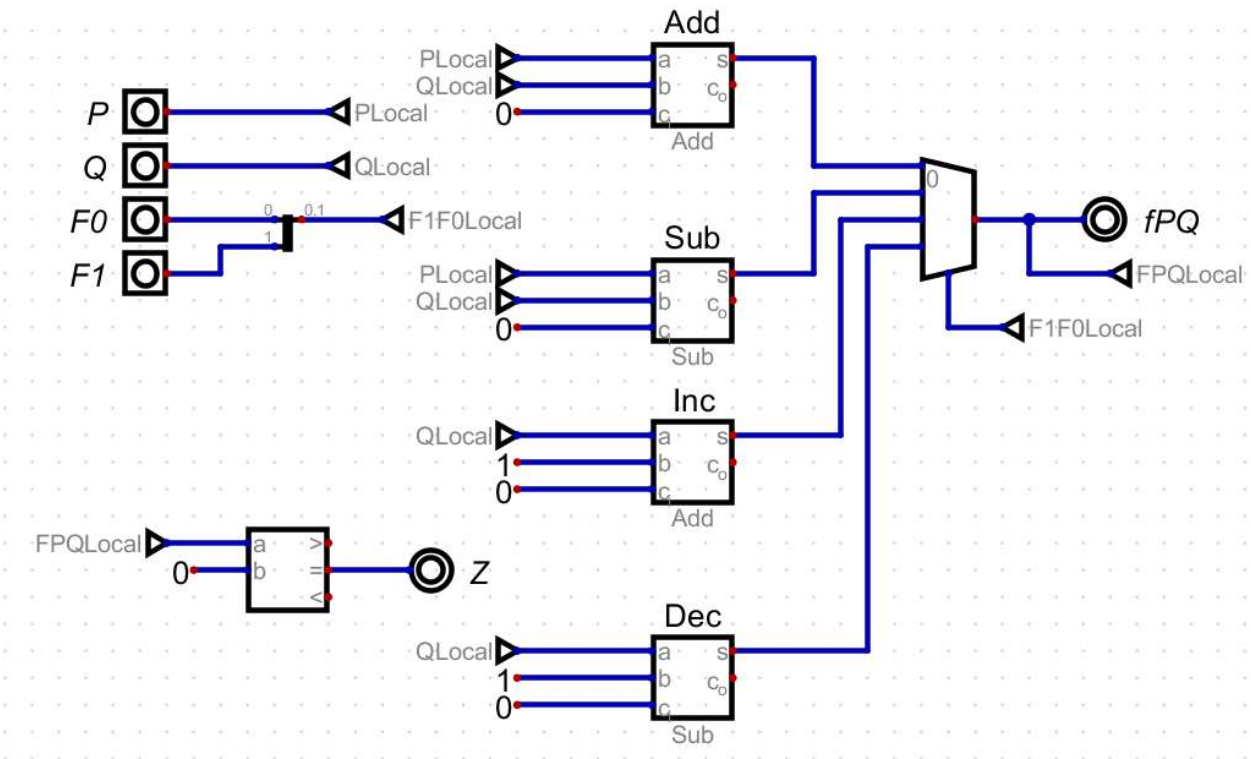

Figure 3: Circuits for ALU

3 Numerical Verification

The simplest way to test this circuit would be to run a program that exercises all the commands. To do so, a program that incorporates all these commands, shown in Figure 4, was executed. In a convoluted mode, this program counts down from the input value halting when it completes counting down to 0.

Memory Address (hex)	Memory Address (decimal)	Data (Hex)	Instruction	Address (Decimal)	Description
00	0	9	**load**	9	Loading the value at address 9 (value 5)
01	1	4A	**add**	10	Adding the value at address 10 (value 1) to address 9 (value 5)
02	2	6B	**sub**	11	Subtracting the value at address 16 (value 2) from address 12 + address 16 (value 5 + 1)
03	3	29	**store**	9	Storing that value back at address 9 (net 1 decrement)
04	4	89	**inc**	9	Incrementing the value at address 9 by 1
05	5	A9	**dec**	9	Decrementing the value at address 9 by 1
06	6	E8	**beq**	8	beq if the value is 0, jump otherwise keep going
07	7	C0	**bra**	0	jump to beginning
08	8	C8	**bra**	8	effective halt
09	9	5	**data**	5	data
0A	10	1	**data**	1	data
0B	11	2	**data**	2	data

Figure 4: Test Program

Shown in Figure 5 is the final state of the program data memory (with the initial state being displayed in Figure 4).

ProgramDataMemory

Address	0	1	2	3
0	9	0x4A	0x6B	0x29
4	0x89	0xA9	0xE8	0xC0
8	0xC8	0	1	2
0xC	0	0	0	0
0x10	0	0	0	0
0x14	0	0	0	0
0x18	0	0	0	0
0x1C	0	0	0	0

Figure 5: Final State of Test Program

4 Summary

The problem requirement for this assignment was to design a Datapath circuit to define the state of the CPU at any time, effectively completing the CPU design. This includes the management of the program management registers including the MAR, MBR, IR, PC, D0, and ALU.

By employing the Datapath diagram displayed in Figure 1, a Datapath circuit was created. The circuit was tested with a countdown program that counts down from the inputted value, memory address 0x09 Figure 4, to 0. As the final state of the program has that address value in Figure 5, 0x09 as zero, this suggests that the design can accomplish the programmed function, exercising all its registers and functions. Therefore, the specified digital circuit design solution satisfies the problem requirements.