Signed Fixed-Point Calculator

Yusuf Husen Khatri, Peter Barkho, Christopher MacKenzie, Eduardo Garcia

ECE 4710/5710 - Computer Hardware Design - Winter 2022

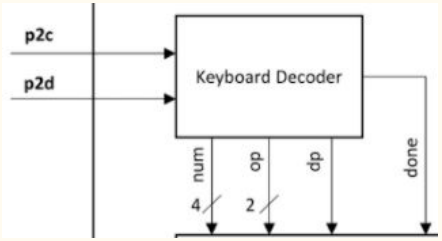
Overview

- Our goal was to create a calculator for signed fixed-point numbers
- The calculator should be able to perform each of the following arithmetic operations
  - Addition, Subtraction, Multiplication, and Division
- The calculator should take input from a keyboard
  - 2 Operands (16-bit Hex), Decimal Points (Fixed-Point), and Operation
- The calculator should output the result of the selected operation on the 7-Segment Display in Hexadecimal format with the proper decimal point position

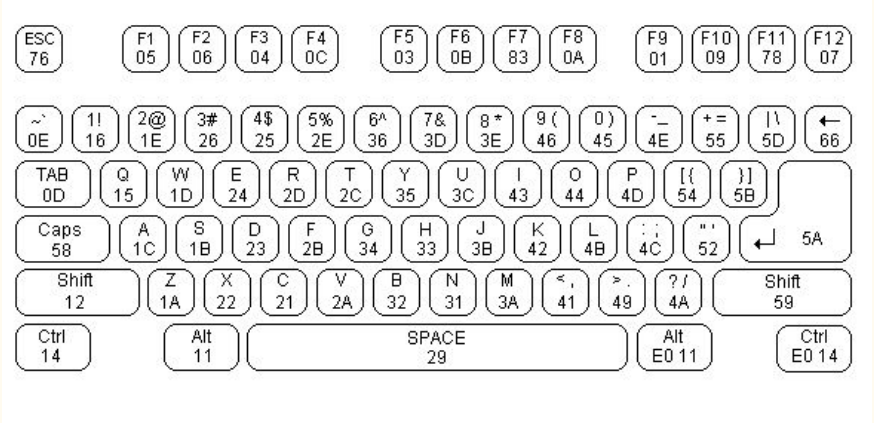
Components

- Keyboard Decoder
- Registers
- Arithmetic Operations Circuit
- 7-Segment Serializer
- Multiplexers
- Finite State Machine

Keyboard Decoder

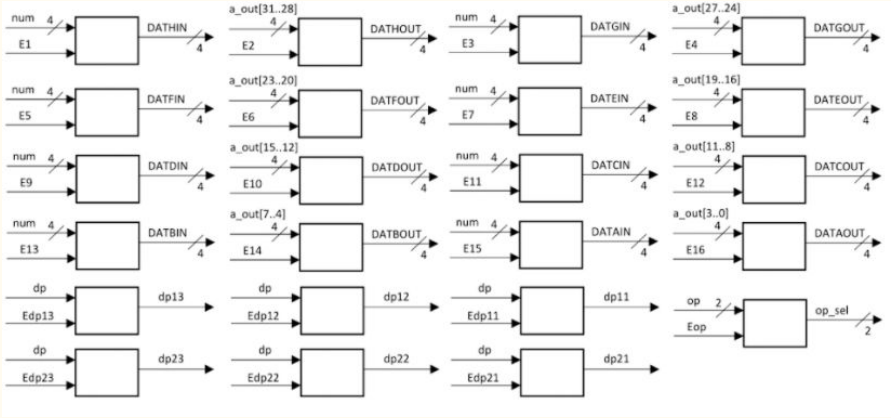


- Uses my_ps2keyboard.vhd component
- Operand Number
  - 0000 - 0 : 0x45
  - 0001 - 1 : 0x16
  - 0010 - 2 : 0x1E
  - 0011 - 3 : 0x26
  - 0100 - 4 : 0x25
  - 0101 - 5 : 0x2E
  - 0110 - 6 : 0x36
  - 0111 - 7 : 0x3D
  - 1000 - 8 : 0x3E
  - 1001 - 9 : 0x46
  - 1010 - A : 0x1C
  - 1011 - B : 0x32
  - 1100 - C : 0x21
  - 1101 - D : 0x23
  - 1110 - E : 0x24
  - 1111 - F : 0x2B
- Operations
  - 00 - Addition (+) : 0x55
  - 01 - Subtraction (-) : 0x4E
  - 10 - Multiplication (x) : 0x22
  - 11 - Division (/) : 0x4A
- Decimal Point
  - 1 - Decimal Point (.) : 0x49



Registers

- We utilize a total of 23 registers in this design (my_rege.vhd)
- The purpose of the registers are as follows:
  - 3 registers to hold decimal point position in operand 1 (1-bit each)
  - 3 registers to hold decimal point position in operand 2 (1-bit each)
  - 8 registers to hold operand 1 and 2 data (4-bits each)
  - 8 registers to hold the calculated output data (4-bits each)
  - 1 register to hold the operation selected (2-bits)
- The registers are enabled by individual enable signals produced by the FSM.
- Timing of the enable signals is based off when inputs are pressed, and when outputs are calculated. Further shown in FSM diagram.

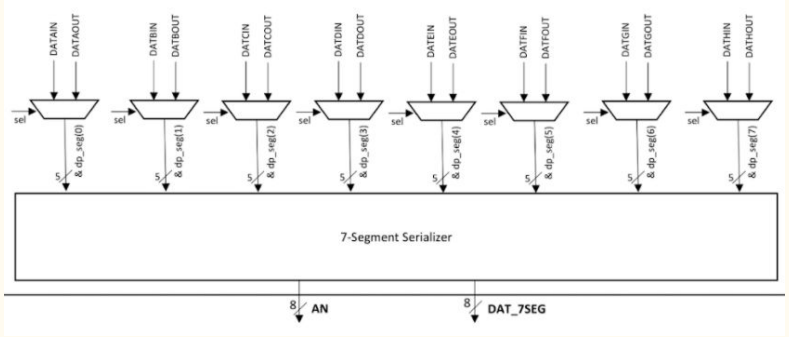


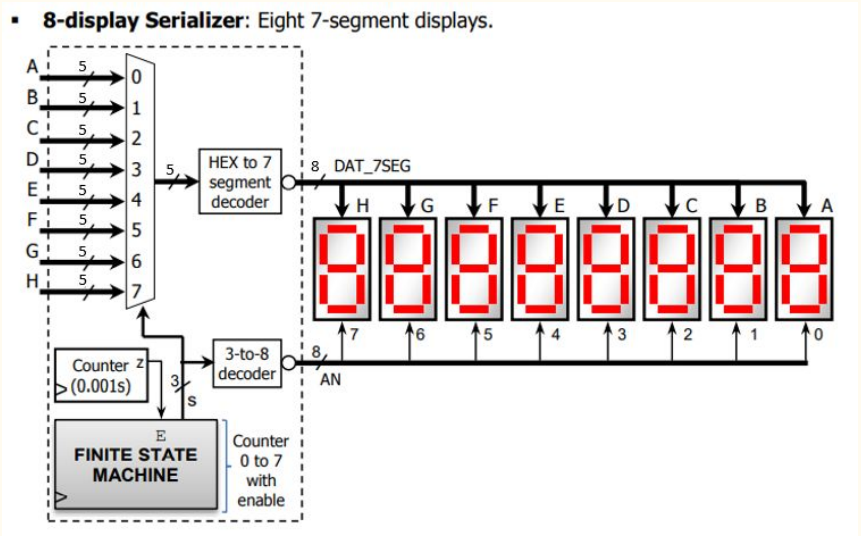
Arithmetic Operations Circuit

- Responsible for performing each of the arithmetic operations
- Mainly composed of 2 Addsubs, a Signed Multiplier, and a Signed Divider
  - Unit 2 Notes
- Addition and Subtraction
  - Performs Alignment (Zero-Padding and Sign-Extension) based on input decimal point position
- Multiplication
  - No need for alignment - Simply performs multiplication with the operands
- Division
  - Performs Alignment and utilizes 4 Precision Bits (Appends “0000” to Operand 1)
- Outputs 32-bit result of selected operation
- Based on the operation and the input decimal point positions, this circuit will also determine the output decimal point position

7-Segment Serializer

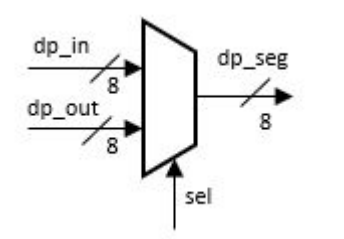
- Based on serializer.vhd and hex2sevenseg.vhd from Lab 3
  - Modified these files for this project to handle the decimal point
- A-H are fed from 8 multiplexers that select between input and output numbers.
- Input numbers shown as keyboard inputs are entered ie when output switch is ‘0’.
- Output numbers are shown when output switch is flipped to ‘1’.
- Output numbers are the calculated values stored in registers.
- The decimal point is handled by combining the multiplex signal into the serializer with the decimal point value. The placement of the point is calculated in the HEX to 7 segment decoder comparative circuit.
- FSM, counter, and 3-8 decoder multiplex display which displays the eight digits and decimal point.

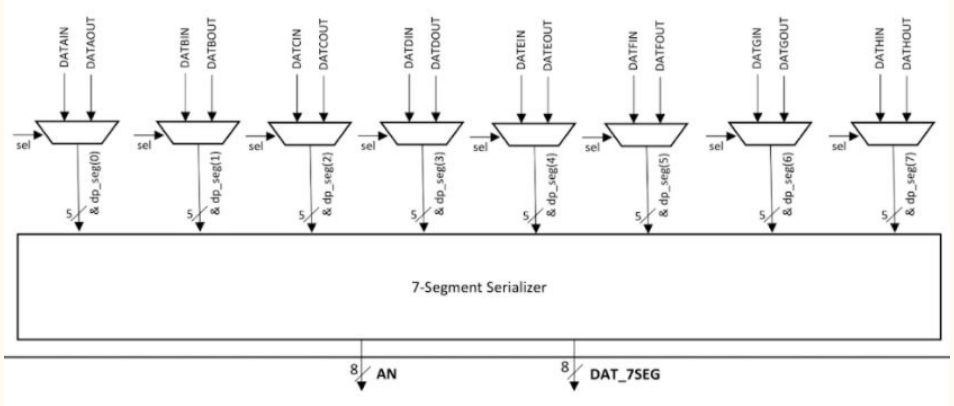




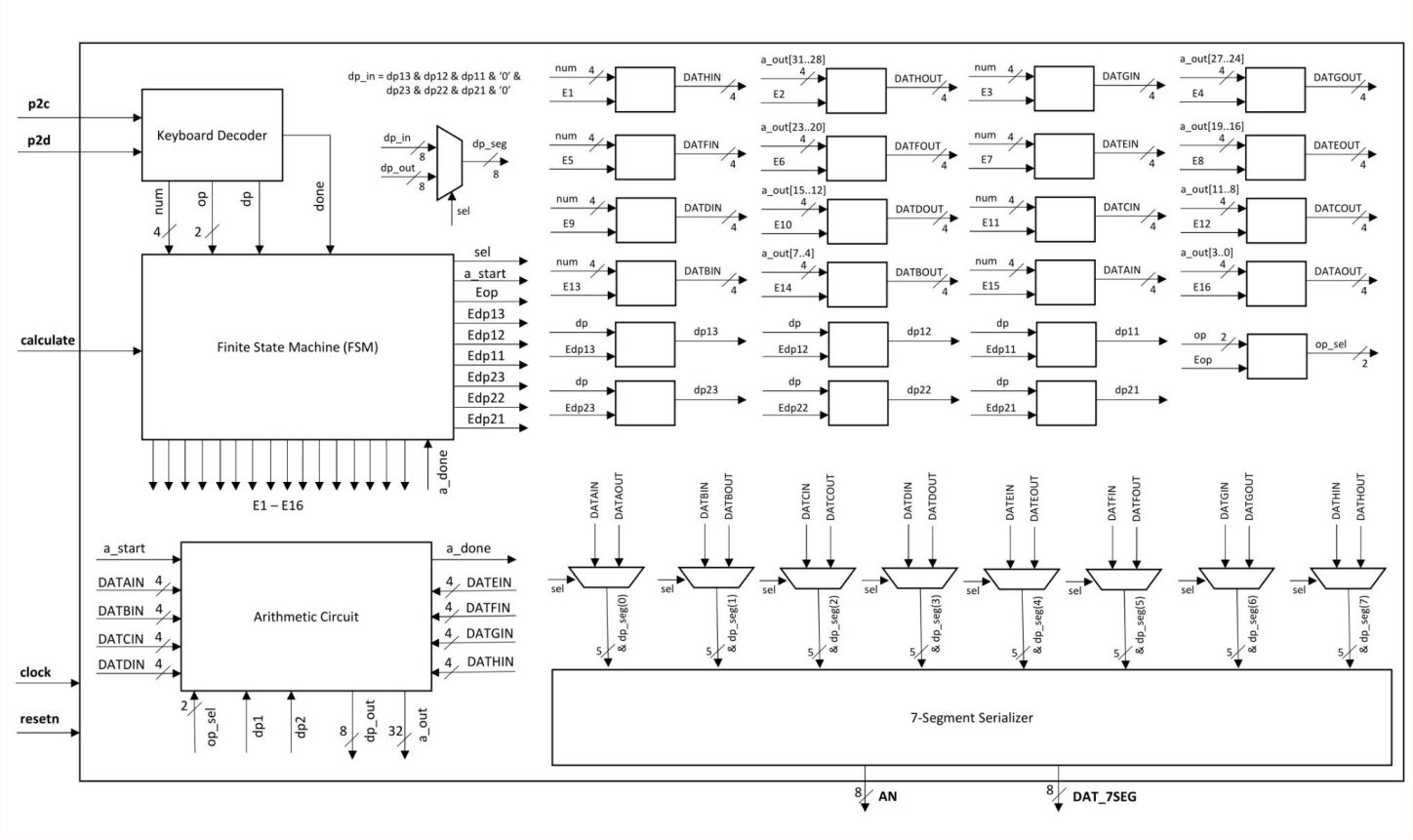
Multiplexers

- As mentioned in the previous slide, the main purpose of the several multiplexers we use in our design was to select between showing the user input on the 7-Segment Display while they are inputting and the arithmetic output on the 7-Segment Display when the calculation is complete

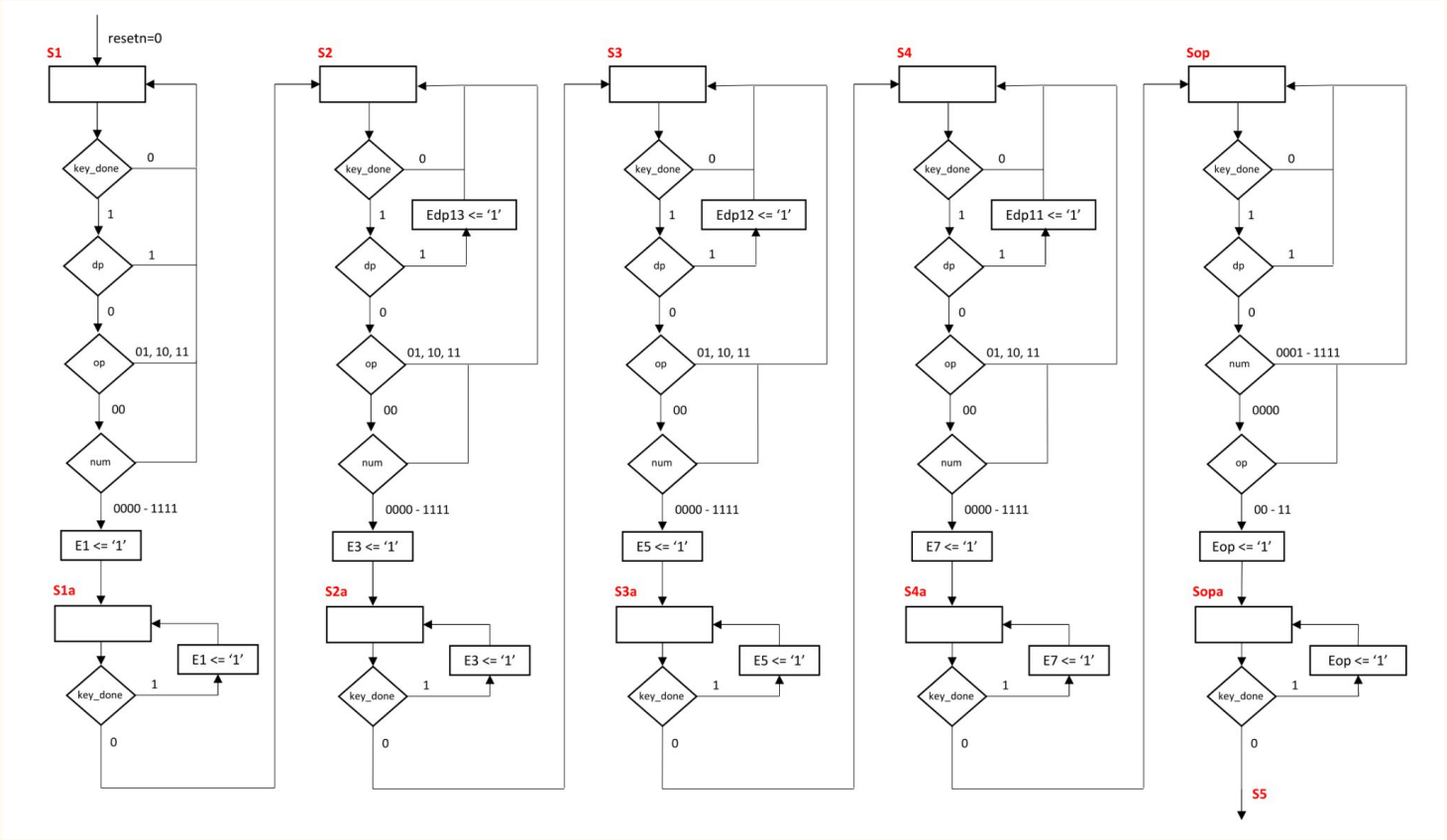




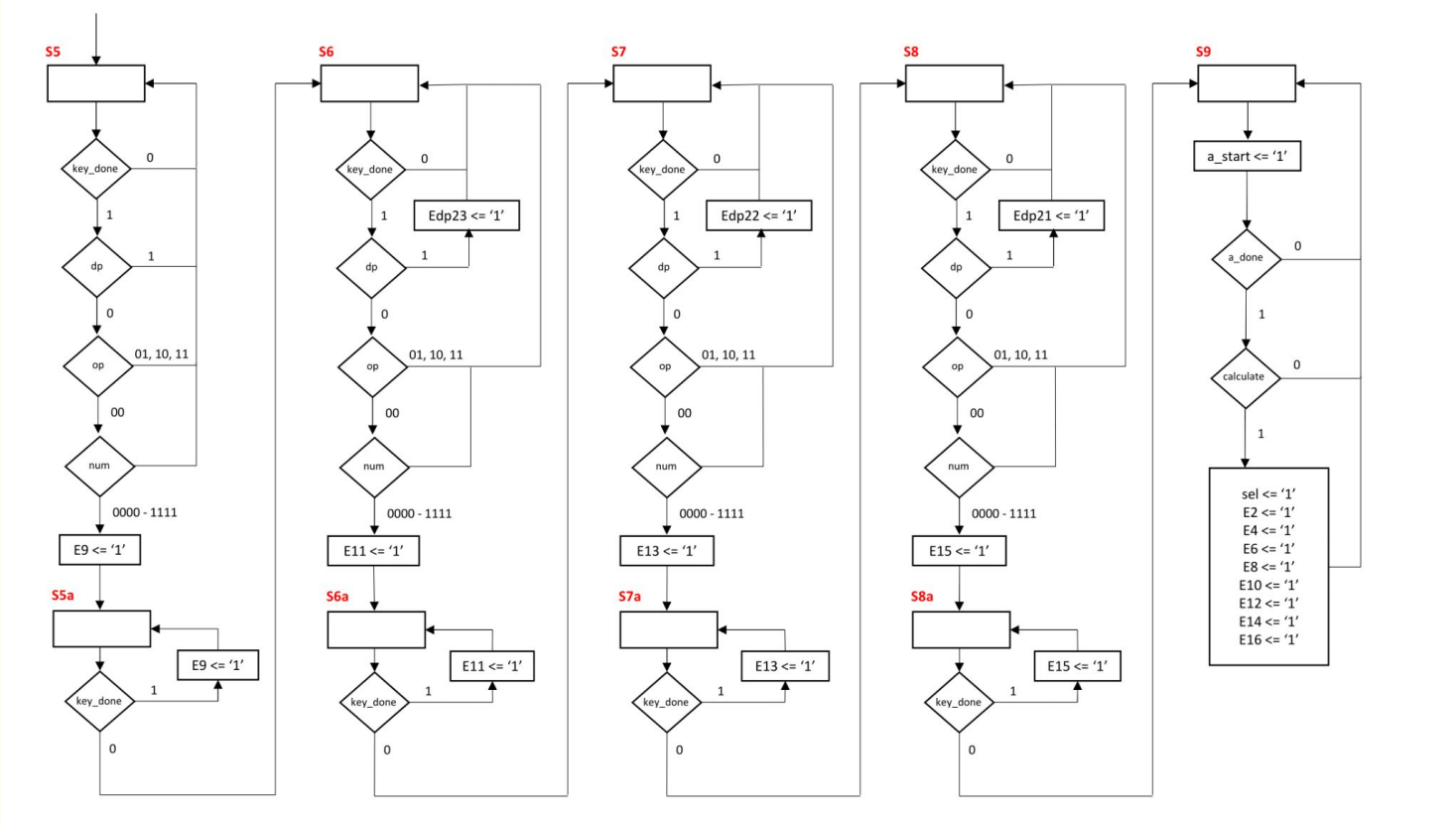
Block Diagram



Finite State Machine



Finite State Machine (cont.)



Project Demonstration

- Addition
  - 37.AB + 1.FC8 = 0039.A780
- Subtraction
  - F.540 - 682.5 = F97D.0400
- Multiplication
  - 3D.21 x 6.CF7 = 1A0.4E6D7
- Division
  - FFE.6 ÷ 000.7 = FFFFFFC.5

- Video Demo (Backup) - https://youtu.be/milBWN9lGGo

Thank you