IA Digital Electronics - Supervision I

Nandor Licker <nl364@cl.cam.ac.uk>

Due noon two days before the supervision

1 Overview

The goal of this exercise is to design an 8-digit calculator capable of adding and subtracting values entered using a 4×4 matrix keypad. The calculator will work with Binary Coded Decimal (BCD) numbers. You are given the high-level diagram of the calculator, indicating its major components and the wires connecting them. Your task is to design the combinatorial or sequential logic implementing each component by deriving the truth tables and the minimal equations, drawing circuit diagrams, and choosing suitable SN74HC series ICs to implement them.
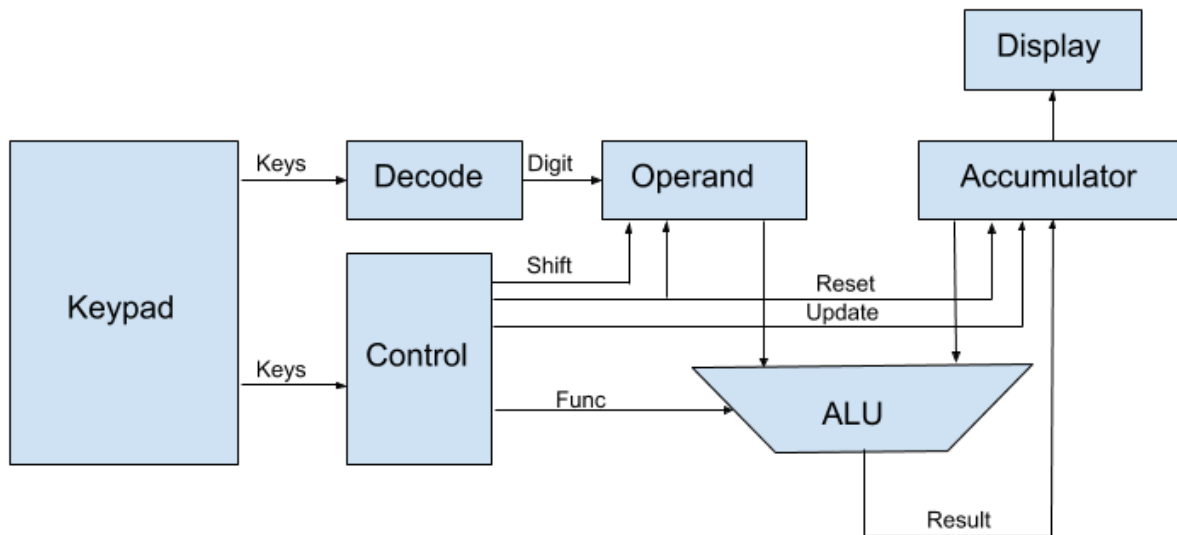

Figure 1: Components and wiring of the 8-digit calculator

Figure 1 outlines the required components:

- 8-digit accumulator

- 8-digit display to display the value in the accumulator

- 8-digit operand register

- Arithmetic Unit to perform addition and subtraction

- 16-key keypad, of which 10 digits and +, - and CLEAR keys are used, arranged in a 4×4 matrix

- A decoder to read a digit from the keypad

- Control logic tying all of them together, deciding when to update the operand, the accumulator and which function to perform in the ALU

Questions

1. How many bits are required to store a decimal digit? How wide are the accumulator and operand registers?

2. On a copy of the diagram, label the width (number of bits) of all signals.

3. Label all sequential components which require a clock signal.

2 Keypad

Values will be entered into the calculator using a matrix keypad, show in Figure 2. Notice that the keypad has 16 keys, but only 8 wires. If each key had its own wire, the number of connections would have been too large, thus keys are arranged in a matrix, where each key is at the intersection of a row and a column. Pressing "1" connects the first row with the first column, pressing "8" connects the 3rd row with the 2nd column. Decoding values is slightly tricky, as we can only power one row at a time, reading the values of the columns: if we power row 1 and keys 5 and E are pressed, we would expect columns 2 and 4 to be high. Your task is to design a circuit to power the rows one by one, decoding which key was pressed by inspecting the outputs. Your circuit should include flip flops for each key, which store a high value if the key was last pressed when its row was inspected. The outputs of the flip flops will be the outputs of this module.
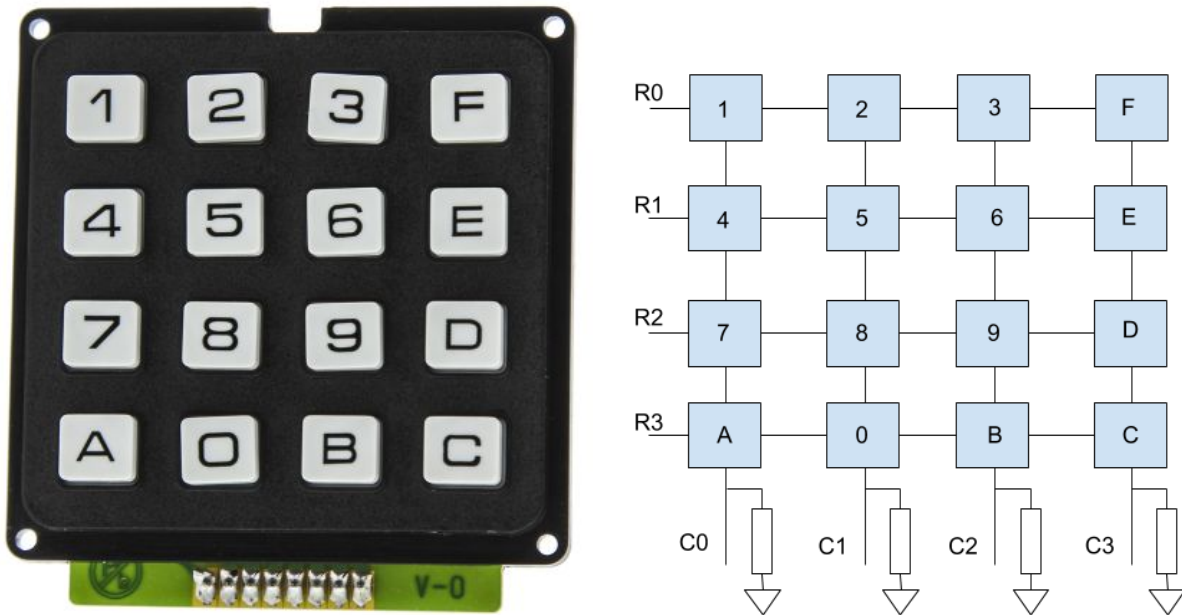

Figure 2: 4×4 matrix keypad

Questions

1. Why are resistors required on the columns? What value should they have (rough estimate - low or high)?

2. Why are resistors not required on the rows, given that we work with SN74HC series ICs?

3. What other passive components might be necessary on the row inputs?

4. Design a circuit with 2 D-type flip-flops to power the rows.

5. Design a circuit with 4 D-type flip-flops to power the rows. What is the common name of this scheme?

6. How would you debounce the input signals?

7. Design a circuit to decode the values of the 16 keys and store the results in flip-flops assigned to each key: draw the truth tables and circuit diagrams. Use your module from Question 3.

8. Is this a Mealy or a Moore machine?

9. How many and what kind of ICs would you need to implement this circuit? Label your circuit diagram to indicate to which IC you would assign each gate. Remembering that the SN74HC series includes multiple-input logic gates, try to minimise the number of ICs.

10. Does your circuit have any functional hazards? How would you eliminate them? Do you need to eliminate hazards at all?

3   Decoder

The keypad module has 16 outputs, however the operand register requires a 4-bit input encoding a digit in binary. Your task is to design a combinatorial circuit to decode the input keys to digits.

Questions

1. Draw the truth tables and derive minimal equations to decode decimal digits.

2. Draw the truth tables and derive minimal equations to decode hexadecimal numbers.

3. How would you construct this module using SN74HC series ICs?

4. Show how you would eliminate any static hazards, if there are any.

4   Operand Register

The numbers entered using the keypad form the operand. We can perform two operations on this register: we can shift the number from the decoder in the place of the units or clear all values to 0. When a digit is pressed, the numbers are shifted left and the digit appears in the place of the units. When the clear button is pressed, all digits are set to 0, clearing the contents of the register.

Questions

1. What are the inputs to this module? What are the outputs?

2. Design a 4-to-1 demultiplexer, drawing the circuit diagram.

3. Design a circuit to store a single digit and to perform the required operations on it.

4. Wire 8 of the units you designed previously to form this module.

5. What value will the register store if you press "1", "2", "3"?

6. What value will the register store if you press "1", "2", "3", "4", "5", "1", "2", "3", "4", "5"?

7. How would you construct this module using SN74HC series ICs?

5   Accumulator

The accumulator is similar to the operand register, except it performs different operations: we can either reset it, similarly to the operand register, or we can load a new value into it.

Questions

1. When do we reset the accumulator?

2. When do we update the accumulator?

3. What are the inputs to this module? What are the outputs?

4. Design a circuit to store a single digit.

5. Wire 8 of the units you designed earlier to form the ALU.

6. How would you construct this module using SN74HC series ICs?

7. Did you construct a Mealy or a Moore machine?

8. Some value combinations stored in the registers are unused - how many D-type flip-flops would you require to store the 8-digit numbers in binary? Given that we display decimal numbers, would it be cost effective to compress the representation of the numbers? What if the calculator was hexadecimal?
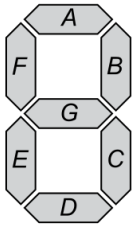
6 Display

Figure 3 shows a 7 segment display - named so because it is controlled by turning on or off each of the 7 segments, labelled A, B, C, D, E, F, and G. The value in the accumulator is displayed using 8 of these displays.

Figure 3: Display

Questions

1. Design a combinatorial circuit to drive the display, assuming inputs are decimal.

2. How many integrated circuits are required for a single digit? For all of them?

3. Design a combinatorial circuit to drive the display, assuming inputs ar hexadecimal.

4. How many wires does the display require? How would you reduce the number of wires to $7 + 8$ and how would you control such a display?

5. Why are hazards not a problem here?

7 Arithmetic Unit

The arithmetic unit takes two operands, the accumulator and the operand register. It either subtracts the operand from the accumulator or adds them together. In this section, we will design the arithmetic unit.

1. Design a half adder.

2. Design a full adder.

3. Design a circuit for subtraction.

4. Design a circuit to add or subtract two digits using the ones you devised earlier.

5. Design a circuit to add or subtract without actually using a subtractor.

6. Assuming a propagation delay of 5ns per gate, what is the propagation delay?

7. How would you connect 8 of the units you designed earlier to form the Arithmetic unit?

8. What is the propagation delay of the entire ALU?

9. How would you implement the ALU using SN74HC series ICs?

8 Control Logic

All that is left is the control logic, tying all the previously designed modules together, generating control signals to operate the operand register, the accumulator and the ALU.

1. Is the control logic stateful? Does the calculator act when you press or release a button?

2. If the user pressed two keys, we probably don't want to do anything - design a circuit to generate a valid flag which is high when exactly one key is pressed.

3. Design a circuit for another flag which is high when a digit is pressed.

4. Draw the truth table incorporating the flags and any state, outputting the required control signals. Describe the behaviour implement by it.

5. Determine the minimal equations for all control signals.

6. How would you implement this control unit using microcode store in EEPROM?

7. Is this a Mealy or a Moore machine?

9 Additional Questions

1. Roughly how many and what kind of ICs do you need to build the calculator? You can use and SN74HC series ICs, including multiplexers, decoders and adders.

2. Assuming an average delay of 10ns for each gate or integrated circuit, estimate the maximal clock frequency. Identify the critical path.

3. How would you generate the clock signal using a 555 IC?