

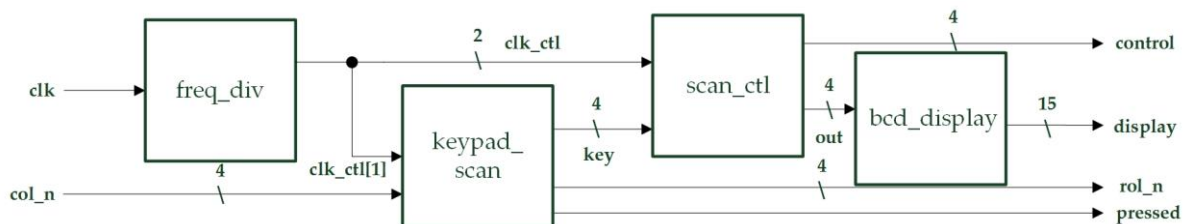
Lab 6: Simple Calculator

103061207 徐安廷 An-Ting Hsu

Design Specification

1. Keypad Scanner

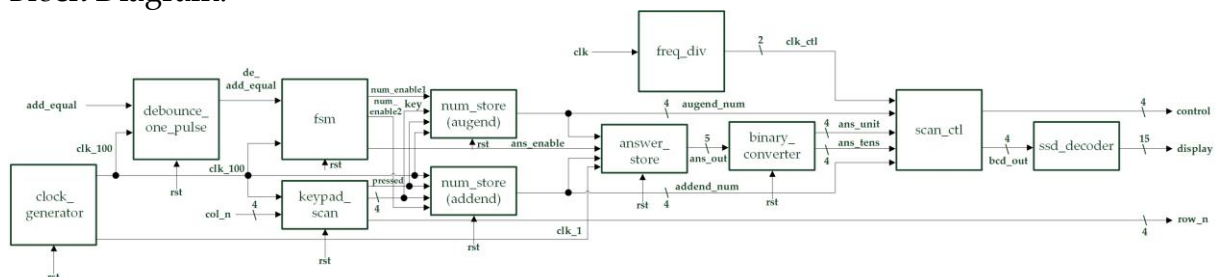
- ✓ Experiment Goal:
Show the letter that was pressed on the 14-segment display.
- ✓ Block Diagram:



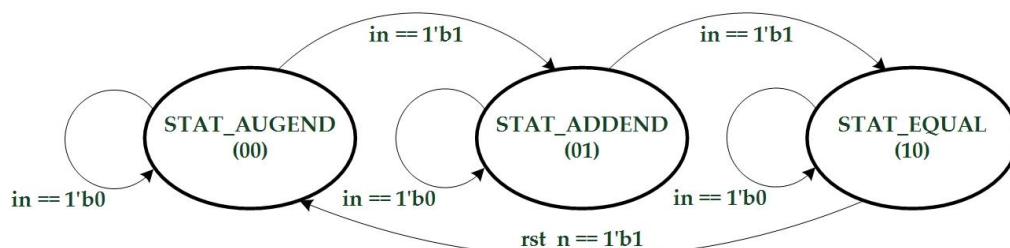
- ✓ I/Os:
Inputs: clk, rst_n, [3:0] col_n.
Outputs: pressed, [3:0] row_n, [3:0] control, [14:0] display.

2. Single Digit Decimal Adder

- ✓ Experiment Goal:
Calculate the sum of one digit augend and addend which is input from keypad, and show the inputs and result on 14-segment display.
- ✓ Block Diagram:



- ✓ State Diagram:



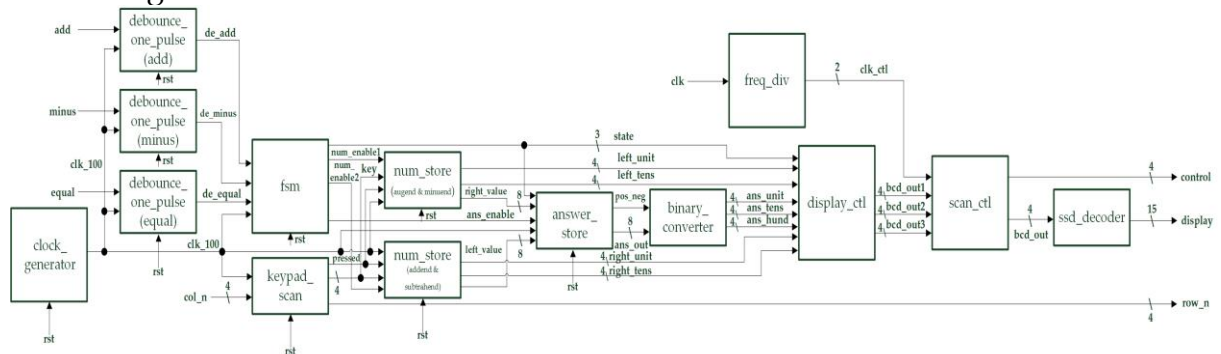
- ✓ I/Os:
Inputs: clk, add_equal, rst_n, [3:0] col_n.
Outputs: [3:0] row_n, [3:0] control, [14:0] display.
- ✓ Details about some module:
 - debounce_one_pulse: Does debounce and one pulse for the input signal.
 - num_store: Stores the number from keypad in a flip-flop.
 - answer_store: Calculates the result of the augend & addend, and stores in a flip-

flop.

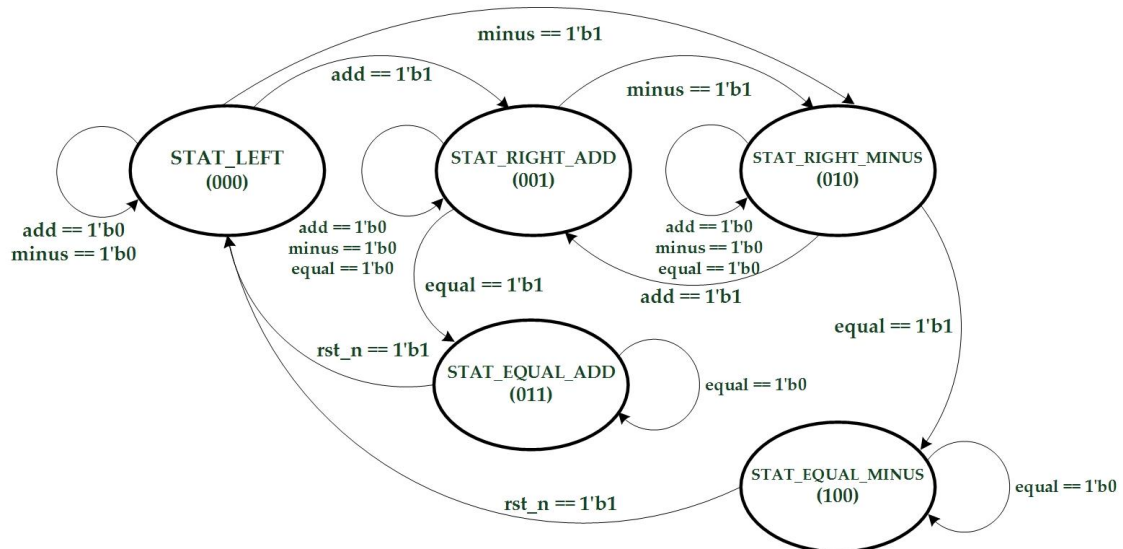
- **binary_converter**: Converts the binary input to BCD codes which is going to show on the 14-segment display.

3. Two Digits Decimal Adder-Subtractor

- ✓ **Experiment Goal**:
Adjust exp 2 to two digits decimal adder and subtractor.
- ✓ **Block Diagram**:



- ✓ **State Diagram**:

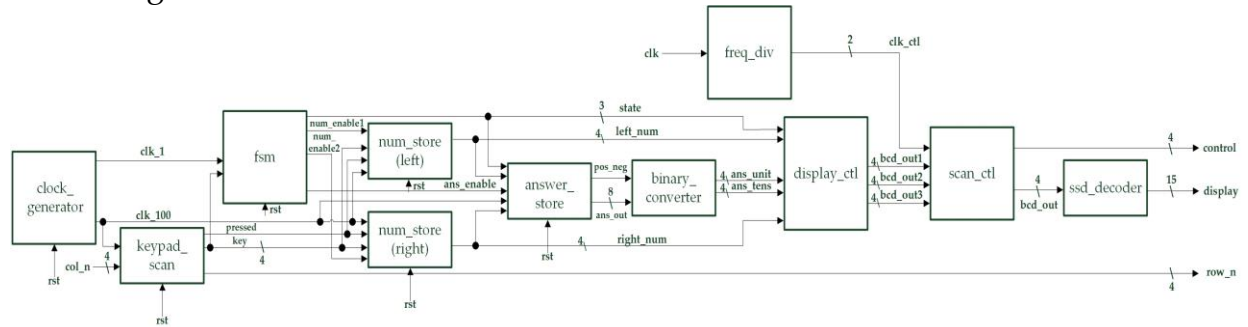


- ✓ **I/Os**:
Inputs: clk, add, minus, equal, rst_n, [3:0] col_n.
Outputs: [3:0] row_n, [3:0] control, [14:0] display.
- ✓ **Details about some module**:
 - **display_ctl**: Determines what is going to show on the 14-segment display according from fsm's state.

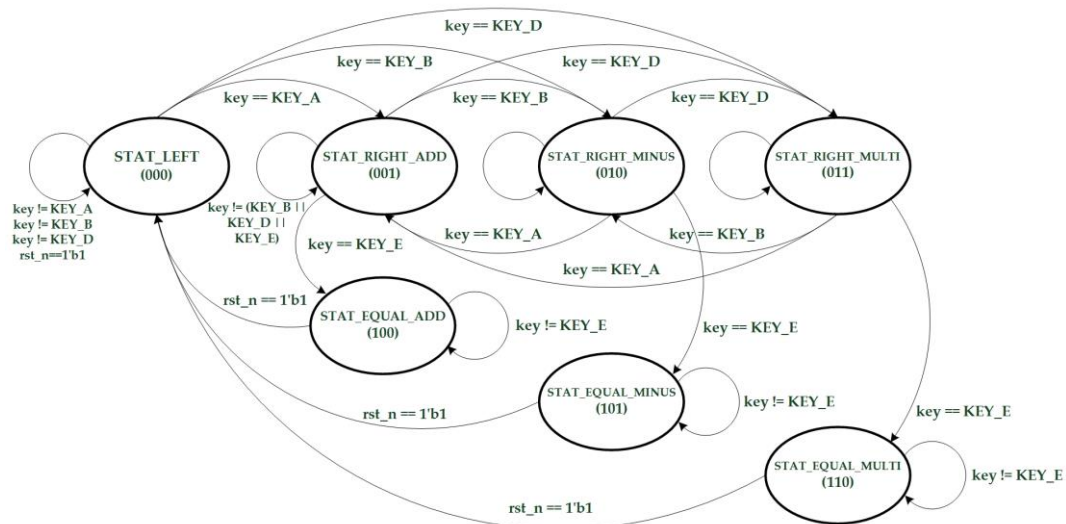
4. (Bonus) One Digit Decimal Adder-Subtractor-Multiplier

- ✓ **Experiment Goal**:
Add subtractor and multiplier function to exp 2. Also, the adder and multiplier core need to be constructed by yourself which means we can not simply write "+", "*" in the Verilog codes.

✓ Block Diagram:



✓ State Diagram:



✓ I/Os:

Inputs: **clk**, **rst_n**, [3:0] **col_n**.

Outputs: [3:0] **row_n**, [3:0] **control**, [14:0] **display**.

Design Implementation

1. Keypad Scanner

- ✓ Copy the keypad scanner code in the handout, and connect the module to frequency divider and display modules to show the key pressed.

- ✓ I/O Pins Assignment:

| | |
|-------------------------------|--|
| clk | R10 |
| rst_n | N3 |
| col_n[0]~col_n[3] | J3, J1, H2, H1 |
| pressed | H5 |
| row_n[0]~row_n[3] | K2, K1, L4, L3 |
| control[0]~control[3] | V8, U8, V6, T6 |
| display[0]~display[14] | U5, T7, R7, V7, V4, T4, T3, R5, N5, R3, U7, T5, V5, N4, P6 |

2. Single Digit Decimal Adder

- ✓ Store the numbers detected from the keypad scanner in module "num_store", and do the calculation in module "answer_store". Pass the result to binary-to-BCD converter

and connect the BCD code output to display modules.

- ✓ I/O Pins Assignment:

| | |
|-------------------------------|--|
| clk | R10 |
| add_equal | N3 |
| rst_n | T2 |
| col_n[0]~col_n[3] | J3, J1, H2, H1 |
| row_n[0]~row_n[3] | K2, K1, L4, L3 |
| control[0]~control[3] | V8, U8, V6, T6 |
| display[0]~display[14] | U5, T7, R7, V7, V4, T4, T3, R5, N5, R3, U7, T5, V5, N4, P6 |

3. Two Digits Decimal Adder-Subtractor

- ✓ Adjust exp2's "num-store" to memorize the two numbers separately in different clock positive edge time. And, add a display controller between the "answer_store" and the display modules which will show different things on different states.
- ✓ I/O Pins Assignment:

| | |
|-------------------------------|--|
| clk | R10 |
| add, minus, equal | P3, P4, N3 |
| rst_n | T2 |
| col_n[0]~col_n[3] | J3, J1, H2, H1 |
| row_n[0]~row_n[3] | K2, K1, L4, L3 |
| control[0]~control[3] | V8, U8, V6, T6 |
| display[0]~display[14] | U5, T7, R7, V7, V4, T4, T3, R5, N5, R3, U7, T5, V5, N4, P6 |

4. (Bonus) One Digit Decimal Adder-Subtractor-Multiplier

- ✓ Add subtractor and multiplier core into exp2's module "answer_store". So, it will calculate the answer by the modules I made.
- ✓ I/O Pins Assignment:

| | |
|-------------------------------|--|
| clk | R10 |
| rst_n | T2 |
| col_n[0]~col_n[3] | J3, J1, H2, H1 |
| row_n[0]~row_n[3] | K2, K1, L4, L3 |
| control[0]~control[3] | V8, U8, V6, T6 |
| display[0]~display[14] | U5, T7, R7, V7, V4, T4, T3, R5, N5, R3, U7, T5, V5, N4, P6 |

Discussion

- ✓ Since teacher gave us two weeks to work on the lab, which means it will need lots of time to write the codes and complete it, and it really took me a few days coding & debugging. Most of the modules have learnt before. The things need to be constructed are some flip-flops which stores and calculates the numbers, the state assignments in finite state machine and display controller of the 14-segment display. Those modules took me most of the time to build it. Though I was once frustrated during the debug process, I finally complete it on time!!!

Conclusion

- ✓ After the lab, I understand how some of the functions in a real calculator works and build it by myself. The calculation may look easy or usual to people not in EE department, but the flip-flops, adders and multipliers are the main core in a contemporary computer. Without these circuits the computer can not do anything for us and the world right now must be definitely in another different sight.

References

1. Keypad Scanner

- ✓ Teaching Handout <Using Keypad> p.6~p.11
→The use of the keypad scanner to detect which button was pressed.

3. Two Digits Decimal Adder-Subtractor

- ✓ iLMS wiki <lab6 bonus documents and example> Binary_to_BCD_Converter.pdf
→Helps me to understand how to convert a binary number into BCD codes and use it in this lab.

4. (Bonus) One Digit Decimal Adder-Subtractor-Multiplier

- ✓ iLMS wiki <lab6 bonus documents and example> Lab03_CombMultiplier.pdf
→Helps me to construct a multiplier by only using adders.