BCD to Binary Converter

By Robert D’Angelo, Joseph Jarbo, & Colton Palmer

Overview

- Take input from a keyboard and use a BCD to binary converter in order to perform operations on the data
- Takes two 3 digit decimal numbers input to the keyboard and adds them together
- Output is displayed on 3 serialized seven segment displays in hexadecimal
- Goal is to demonstrate the ability to take input from a keyboard and convert it to something that can be utilized in a digital circuit

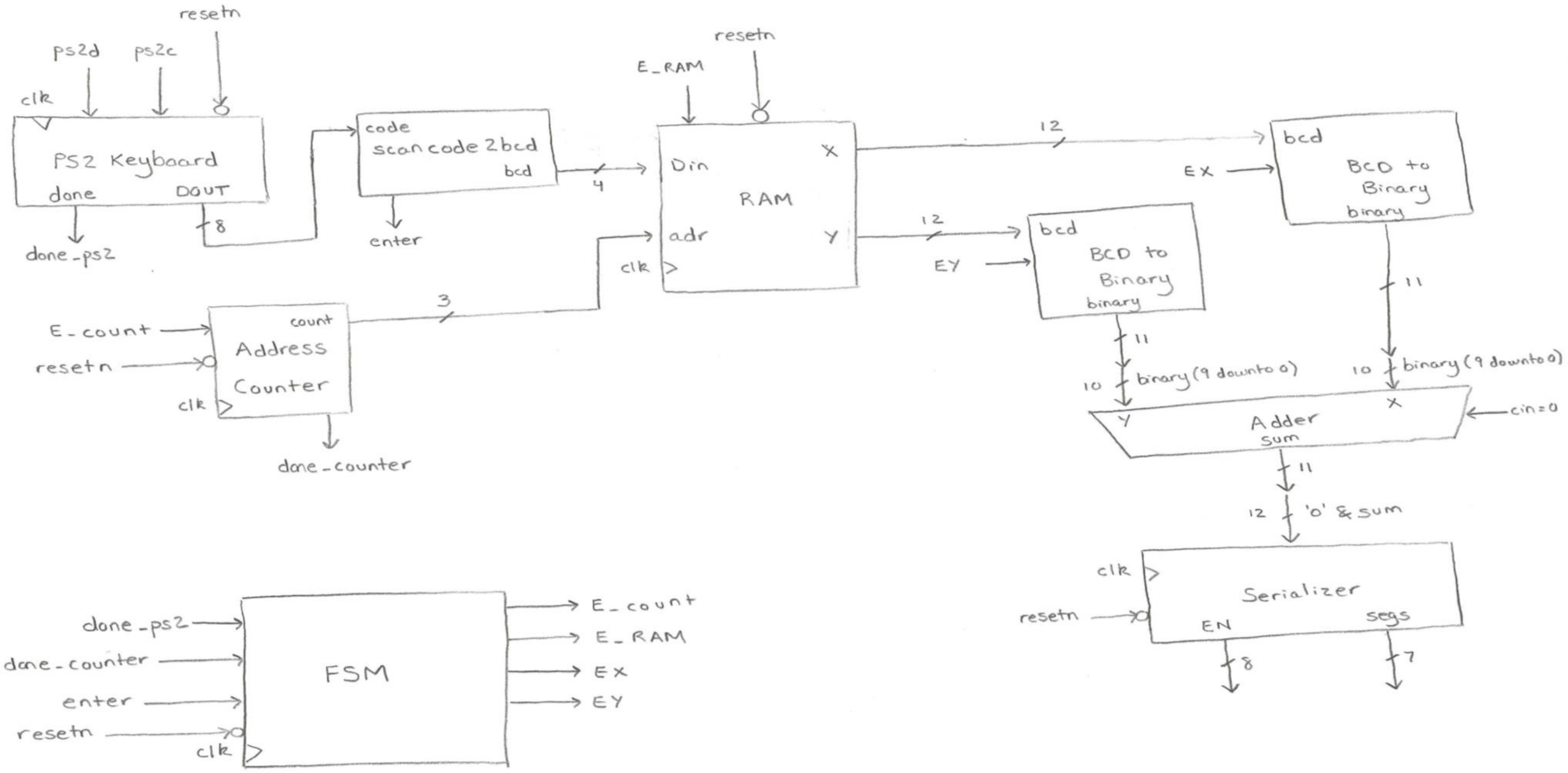
BCD to Binary Algorithm

- 12 bit input is split into groups of four (nibbles)
- Each nibble represents a place value in the decimal system (ones, tens, hundreds)
- To convert, the nibble in the ones place is multiplied by “01”, tens place is multiplied by “1010”, and hundreds place is multiplied by “1100100”
- The three products are added together to obtain the result in binary
- These operations are possible with VHDL’s ieee.std_logic_unsigned.all library

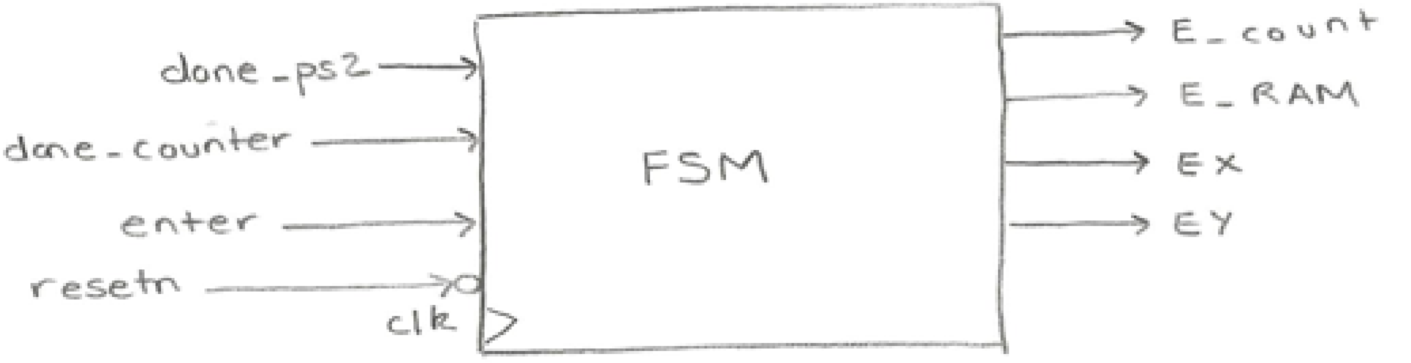
Example

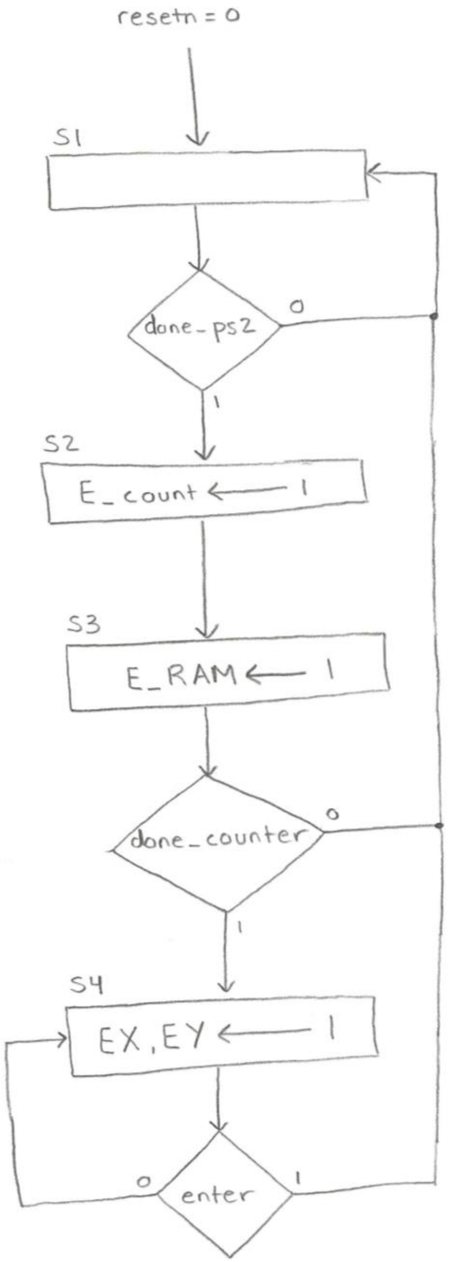
- Input: 123+101
- 123 in BCD: 0001 0010 0011 101 in BCD: 0001 0000 0001
- 123 conversion: (“0011” * “01”) + (“0010” * “1010”) + (“0001” * “1100100”)
- **Result:** 123 = “0111 1011” (binary)
- 101 conversion: (“0001” * “01”) + (“0000” * “1010”) + (“0001” * “1100100”)
- **Result:** 101 = “0110 0101” (binary)
- Sum of Results: “1110 0000”
- In Hexadecimal: 0E0 (displayed on board)

Block Diagram



Finite State Machine





Test Cases

- 132 + 201 = 333 (decimal) = 14D (hex)
- 129 + 275 = 404 (decimal) = 194 (hex)
- 345 + 213 = 558 (decimal) = 22E (hex)
- 999 + 999 = 1,998 (decimal) = 7CE (hex)

Potential Design Improvements

- Use sequential logic for BCD to binary conversion if more sequential components after the converter
- Add functionality to allow for larger numbers to be input
- Allow the user to select how the output is displayed (decimal or hex)

Questions?