CS 51 Homework 5

Jessie Li

October 20, 2024

1.

See `q1.ys`.

My implementation for inverting alphabetic characters is based on the observation that the lowercase and uppercase (binary) ASCII representations of a letter differ only at bit 6, corresponding to $2^5 = 32$. Bit 6 is 0 in uppercase and 1 in lowercase, since each lowercase letter is 32 greater than its uppercase counterpart. Thus, once a character x is determined to be alphabetic, we can invert its case by flipping bit 6 with x XOR 32. To determine whether a character is alphabetic, I set bit 6 to 0, converting to uppercase, and check whether the result is in the range [A, Z].

Testing

- Uppercase letters, lowercase letters, boundary letters (A, Z, a, z)
- Non-alphabetic characters < A, > z, between Z and a
- Sequences of alphabetic and non-alphabetic characters
- Example sequence that tests all of the above: `.!3AJZ$]aez{` $\rightarrow$ `.!3ajz$]AEZ{`

2.

Notes on Behavior

- `RAMuse` must be 1 for any read/write
- If reading from an empty KB, `data out` = `0x00`
- `KB RdEn` = 1 only if:
  - `RAMuse` = 1, `RAMread` = 1, `RAMaddr` = `0x00fffe04`
  - In my circuit, checking `RAMaddr` ensures that characters in a non-empty KB are consumed on clock rise only when reading from the KBDR (and not some other readable register, like KBSR).

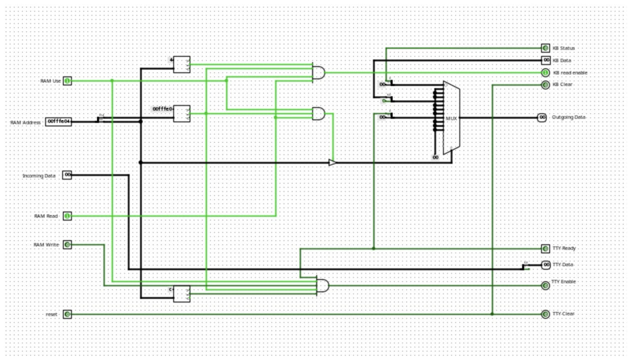


Figure 1: I/O box

- `TTY Enable` = 1 only if:
  - `RAMuse` = 1, `RAMwrite` = 1, `TTY Ready` = 1, `RAMaddr` = `0x00fffe08`
  - Since my `data in` is directly linked to `TTY Data`, the circuitry that sets the value of `TTY Enable` is responsible for checking whether the conditions are satisfied for writing to the display.
- Any unspecified bit is 0 for readable registers KBSR (`0x00fffe00`), KBDR (`0x00fffe04`), and DSR (`0x00fffe08`), so `data out` = `0x00` when reading padding bytes for any of these. For DDR (`0x00fffe0c`), `data out` floats for all 4 bytes because I assume that the display is only written to and not read from.
- None of the registers are both readable and writable, so no error occurs when `RAMread` = 1 and `RAMwrite` = 1. Whether the circuit reads, writes, or does nothing will depend on other inputs, particularly `RAMaddr`.

Additional Testing

In addition to the behaviors described above, I tested/observed:

- Single and consecutive reads from KBDR and writes to TTY
- `data out` for reads of KBSR `0x00fffe00` is 1 when KB contains characters, 0 otherwise
- `data out` floats and `TTY Enable` = 0 if `RAMuse` = 0 or invalid `RAMaddr` (ex. `0x0000cafe`)
- If the KB is non-empty while reading from some other register, say KBSR, the contents of KB are unchanged after clocking
- Asserting `reset` clears both KB and TTY

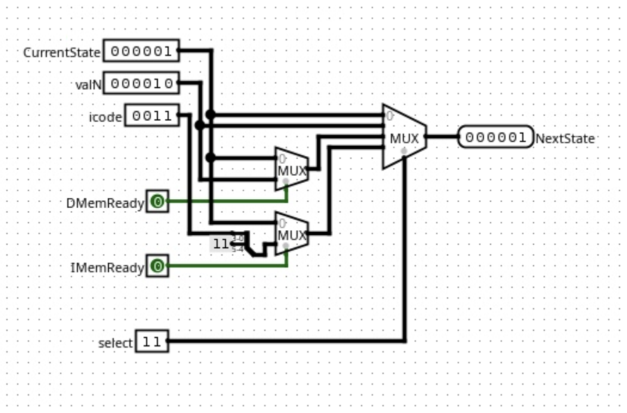
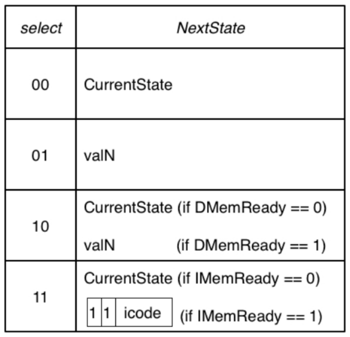


Figure 2: Microsequencer

3.

Testing

I fixed `CurrentState`, `valN`, and `icode` and varied `select`, `DMemReady`, and `IMemReady` to test each of the cases provided in the assignment:



select	*NextState*
00	CurrentState
01	valN
10	CurrentState (if DMemReady == 0) valN (if DMemReady == 1)
11	CurrentState (if IMemReady == 0) 1 1 icode (if IMemReady == 1)

For example, if `CurrentState` = 0000001, `valN` = 0000010, and `icode` = 0011,

select	DMemReady	IMemReady	NextState
00	any	any	0000001
01	any	any	0000010
10	0	any	0000001
10	1	any	0000010
11	any	0	0000001
11	any	1	110011