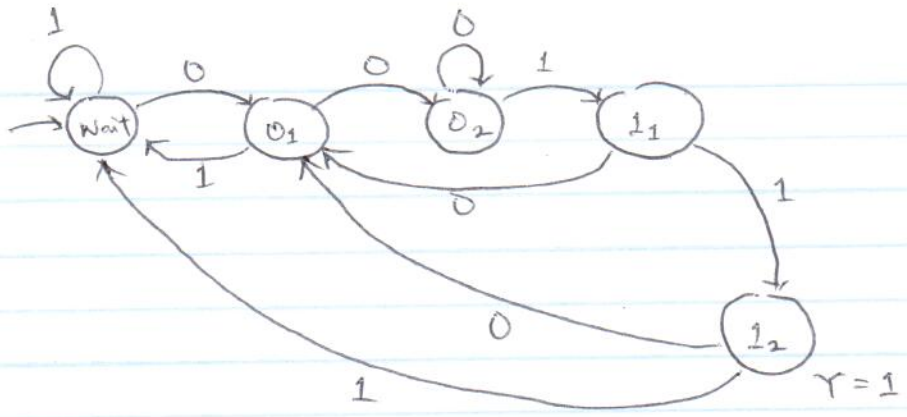
Fsm to detect 0011, output Y = 1



Verilog example

RTL Design

RTL design is a design abstraction that models a synchronous digital circuit in terms of the flow of data between hardware registers and the logical operations performed on them.

HLSMs help us describe behaviors that are too complex for FSMs. Extend FSMs with
- Multi-bit inputs/outputs
- Local storage
- Arithmetic operations

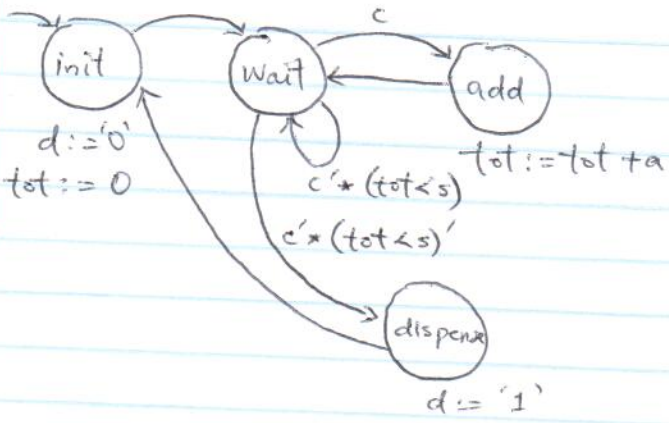
HLSM conventions:
- Single bit: '0' (single quotes)
- Integer: 0 (no quotes)
- Multi-bit: "0000" (double quotes)
- == for equal ; := for assignment
- // for comments
- Multi-bit outputs must be registered through local storage.

HLSM to Datapath example

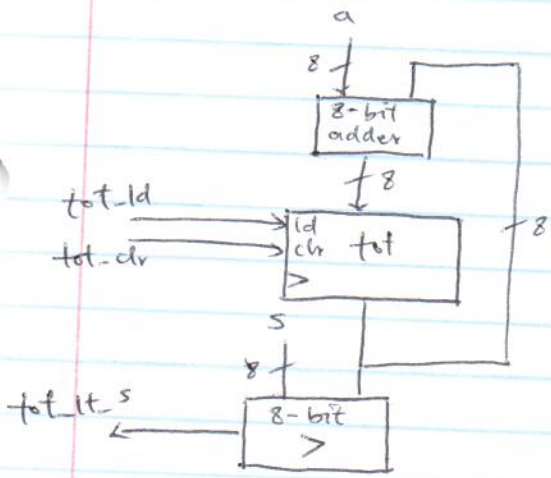
Soda dispenser

Inputs: c (bit), a (8 bits), s (8 bits)
Output: d (bit)
Local storage: tot (8 bits)

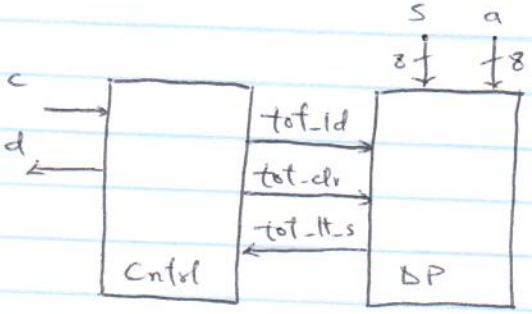
Step 1: HLSM



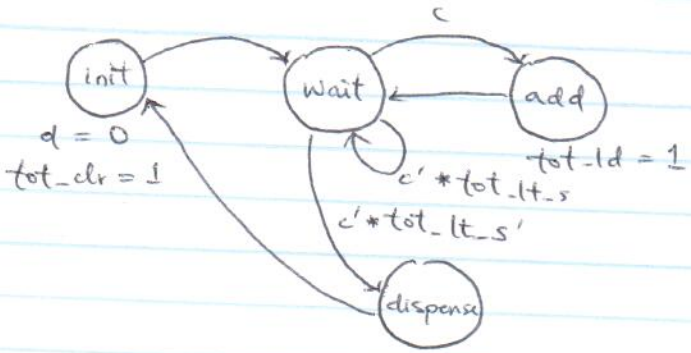
Step 2: Datapath



Step 3: Controller + DP



Step 4: Controller FSM



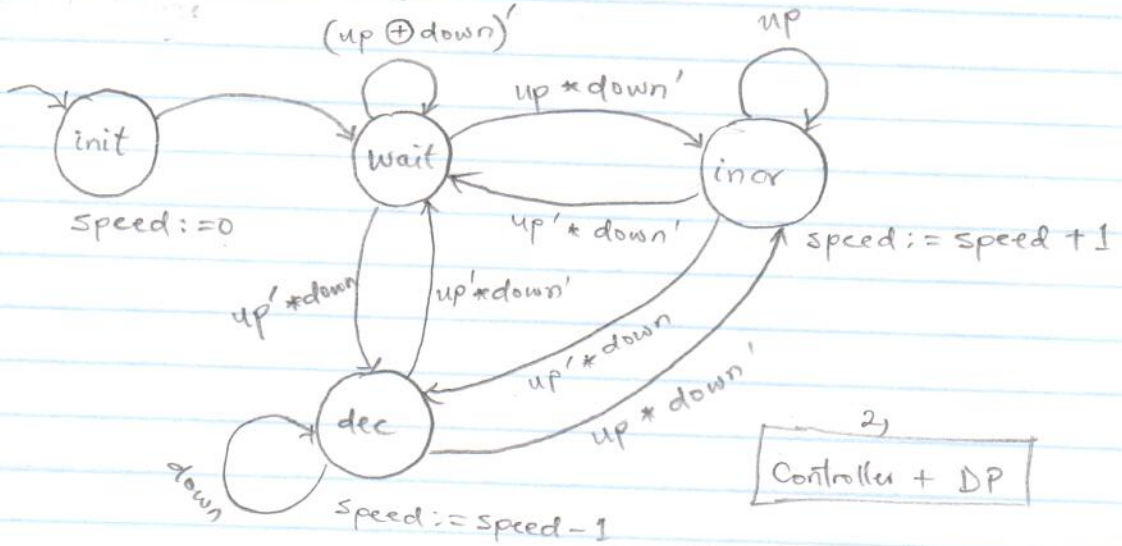
Treadmill Speed Controller

A system to control the speed of the conveyor belt on a treadmill

- Speed is a 4-bit value that is controlled by two buttons
  - Up button increases speed by one
  - Down button decreases speed by one
  - If both are pushed, no change in speed
- Speed must initialize to zero upon startup

1) Capture HLSM

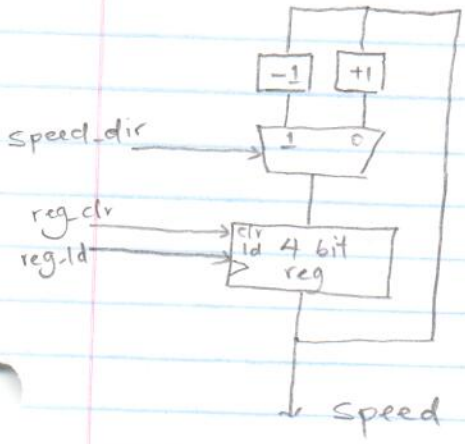
Inputs: up(bit), down(bit)
Outputs: speed (4 bits reg)



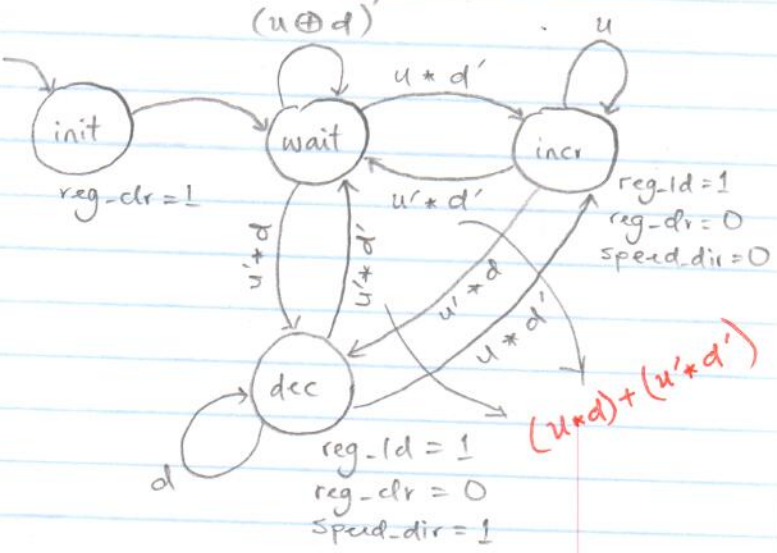
2)

Controller + DP

3) DP

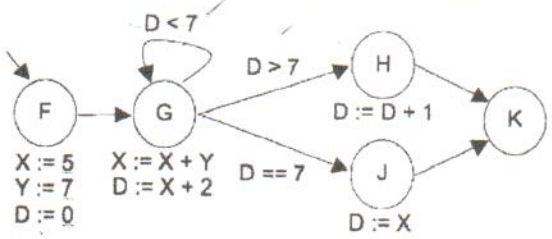


4) Controller FSM



Sample Exam question!

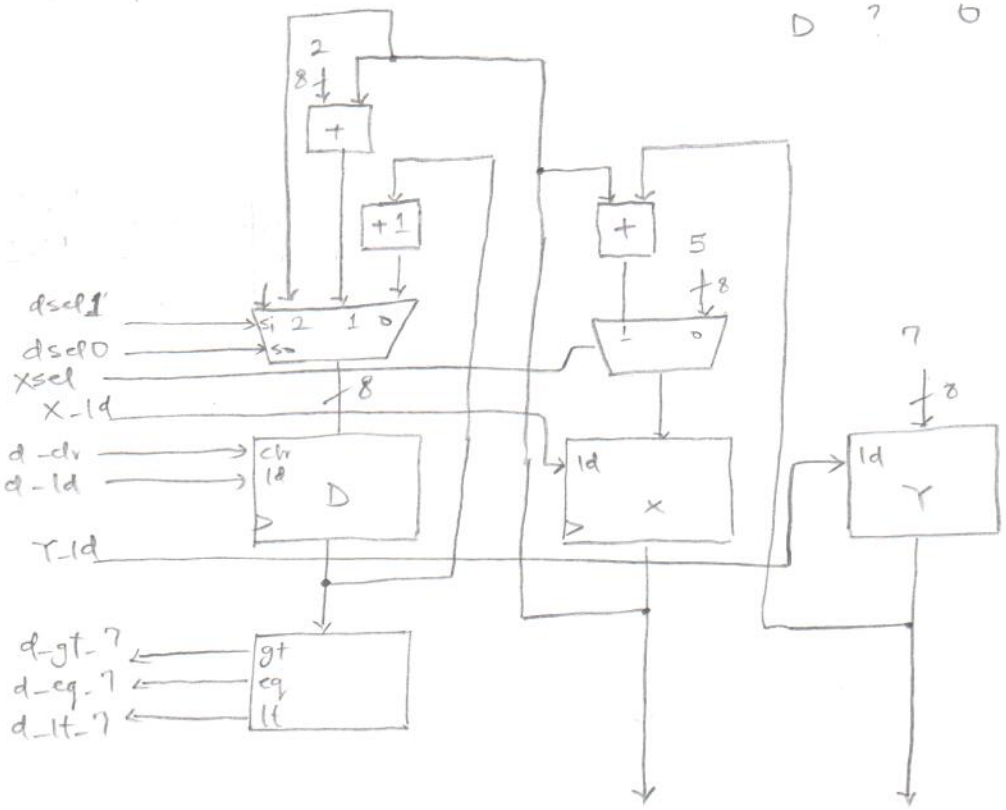
Outputs: X (8-bits), Y (8-bits)
Local Storage: X (8-bits), Y (8-bits), D (8-bits)



Value of D in K? 19

	F	G	G	J	K
X	?	5	12	19	19
Y	?	7	7	7	7
D	?	0	7	14	19

Datapath



Controller FSM