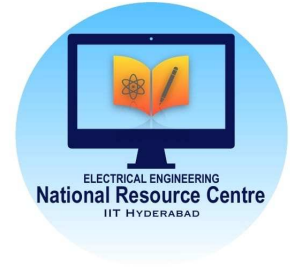




Finite State Machine



G V V Sharma*

CONTENTS

1	The Decade Counter	1
2	Finite State Machine	1
	References	2

Abstract—This manual explains state machines by de-constructing a decade counter.

1 THE DECADE COUNTER

The block diagram of a decade counter (repeatedly counts up from 0 to 9) is available in Fig. 0. The *incrementing* decoder and *display* decoder are part of *combinational* logic, while the *delay* is part of *sequential* logic.

2 FINITE STATE MACHINE

- Fig. 1 shows a *finite state machine* (FSM) diagram for the decade counter in Fig. 0. s_0 is the state when the input to the incrementing decoder is 0. The *state transition table* for the FSM is Table 0 in [1] where the present state is denoted by the variables W, X, Y, Z and the next state by A, B, C, D .
- The FSM implementation is available in Fig. 2. The *flip-flops* hold the input for the time that is given by the *clock*. This is nothing but the implementation of the *Delay* block in Fig. 0.
- The hardware cost of the system is given by

$$\text{No. of D Flip-Flops} = \lceil \log_2 (\text{No. of States}) \rceil \quad (2.1)$$

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

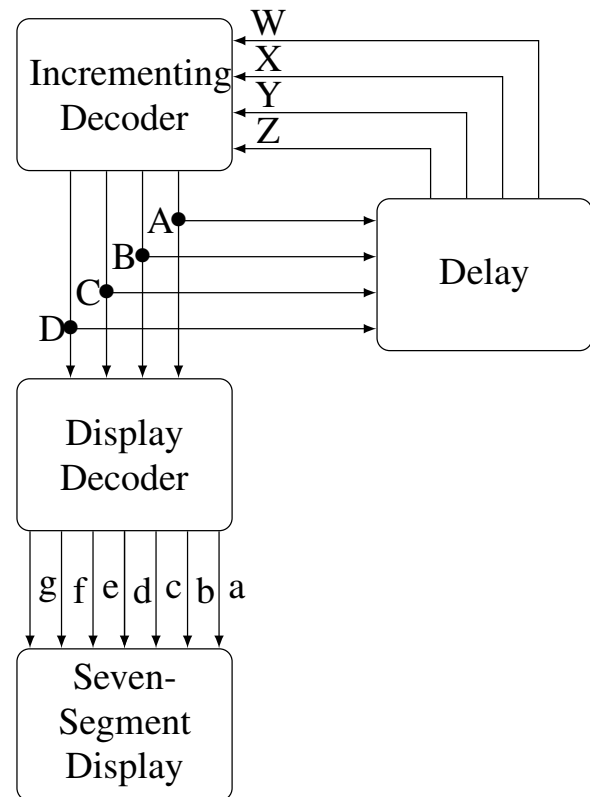


Fig. 0: The decade counter

- For the FSM in Fig. 1, the number of states is 9, hence the number flipflops required = 4.
- Draw the state transition diagram for a decade down counter (counts from 9 to 0 repeatedly) using an FSM.
- Write the state transition table for the down counter.
- Obtain the state transition equations with and without don't cares.
- Verify your design using an arduino.
- Repeat the above exercises by designing a circuit

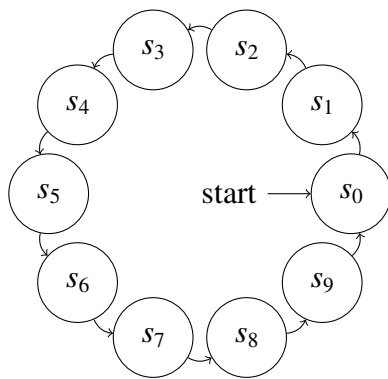


Fig. 1: FSM for the decade counter.

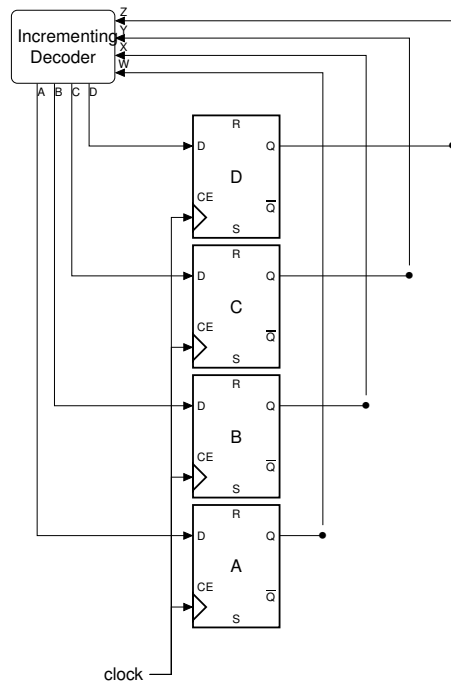


Fig. 2: Decade counter FSM implementation using D-Flip Flops.

that can detect 3 consecutive 1s in a bitstream.

REFERENCES

- [1] G. V. V. Sharma. Karnaugh Map. [Online]. Available: https://github.com/gadepall/arduino/raw/master/ide/kmap/gvv_kmap.pdf