

Day 2 Notes

Zach Neveu

January 13, 2020

State Machine Design Review: "01" string recognizer

Table 1: Flip Flop Equations

x	PS (state Q)	NS (next state Q*)	Z
0	0	1	0
0	1	1	0
1	0	0	0
1	1	0	1

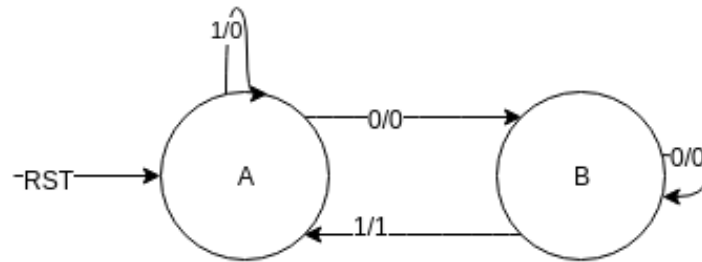
Lecture Notes

- Linux on Pynq: half Ubuntu, half Petalinux
- Should have most common packages
- Petalinux has BSP for Zynq
- Kernel has drivers for AXI busses etc. and writing to PL
- Overlay: a hardware library to be written to the PL

Overlays

- Programmable FPGA IP core
- IP bitstream
- C code to expose functionality
- Python wrapper for c code
- Base overlay: access to headers, buttons, switches, leds, PMODs, audio etc.
- Base overlay designed in Vivado (components reusable for other designs)
- Base overlay has Microblaze for each PMOD interface and one for Arduino header
- Overlay usually includes: bitstream, TCL script to determine IP, Python API

"01" string recognizer



Input: 0, 0, 0, 0, 1, 1
 State: A, B, B, B, A, A
 Output(Z): 0, 0, 0, 0, 1, 0

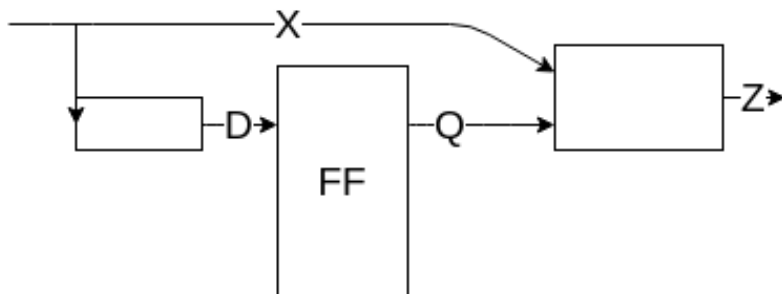


Figure 1: State Machine Example