

Recitation 4: Digital Logic and Shift Registers

I. Basic Digital Logic Overview

Logic gates are the fundamental building blocks of digital logic.

AND Gate



A	B	C
0	0	0
1	0	0
0	1	0
1	1	1

OR Gate



A	B	C
0	0	0
1	0	1
0	1	1
1	1	1

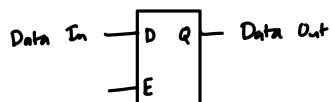
NOT Gate



A	B
1	0
0	1

II. Shift Registers

D Latch

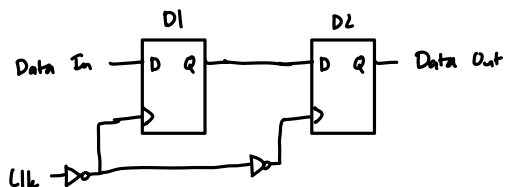


- Enable is HIGH: Q will take the value of D
- Enable is LOW: Q will hold its value

E	D	Q
0	0	Q
0	1	Q
1	0	0
1	1	1

D Flip-Flop

- A pair of D latches

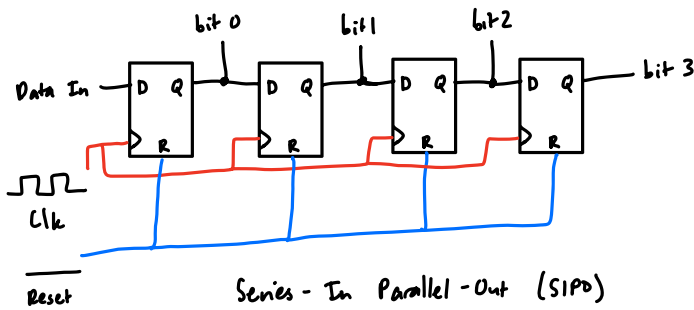


- When Clk is 0, the output of D1 takes the D input and the output of D2 doesn't change
- When Clk is 1, the previous output of D1 is saved to D2

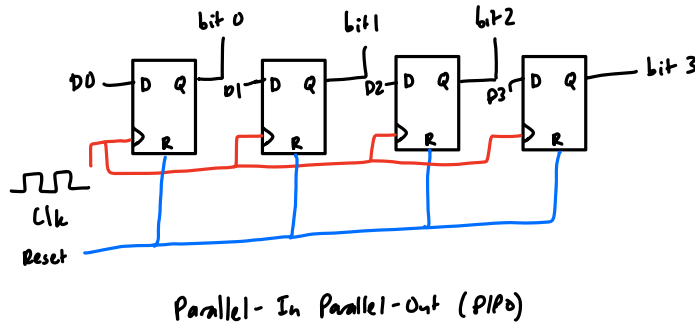
} data out changes on the clock rising edge

└ prevents Data Out from potentially taking on multiple values during a single clock cycle

Shift Register

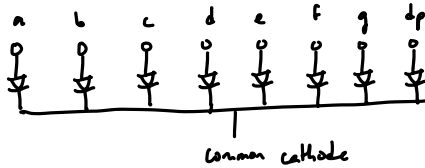


- every clock cycle, a new bit enters the first flip-flop and the bits stored in the other flip-flops are shifted
- one data pin + one clock pin = many outputs



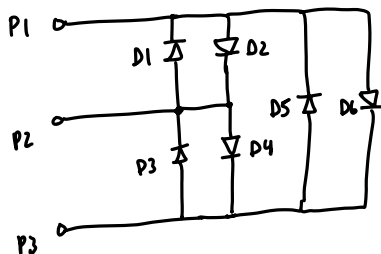
- basically a buffer
- data storage / delay

III. 7-Segment Display Overview



Digit	a	b	c	d	e	f	g
0	1	1	1	1	1	1	0
1	0	1	1	0	0	0	0
⋮							

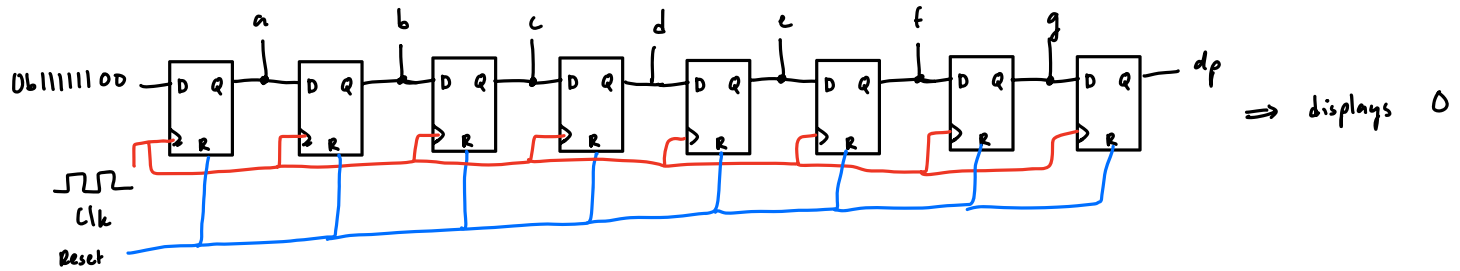
- Individual addressing
 - each LED has its own control line (simple but inefficient)
- Multiplexing
 - a-g segment lines are shared across digits
 - one digit enabled at a time
 - switch quickly between digits
- Charlieplexing
 - utilizes "tri-state" nature of MCU pins (output HIGH and LOW, input high-impedance)



P1	P2	P3	D1	D2	D3	D4	D5	D6
1	z	0						X
1	0	z		X				
0	1	z	X					
z	1	0				X		
0	z	1					X	
z	0	1			X			

III. Shift-Register Addressing of 7-Segment Display

- each LED segment is connected to the output of a SISO shift register
- on the rising edge of each clock cycle, we shift in a byte, where each bit corresponds to the on/off state of a segment



SN74ALS95 Shift Register (ours)

SISO cascaded with PIPD

- data is shifted into SISO register without affecting display
- once all the bits are loaded, the data is transferred to the PIPD register
- display updates simultaneously

