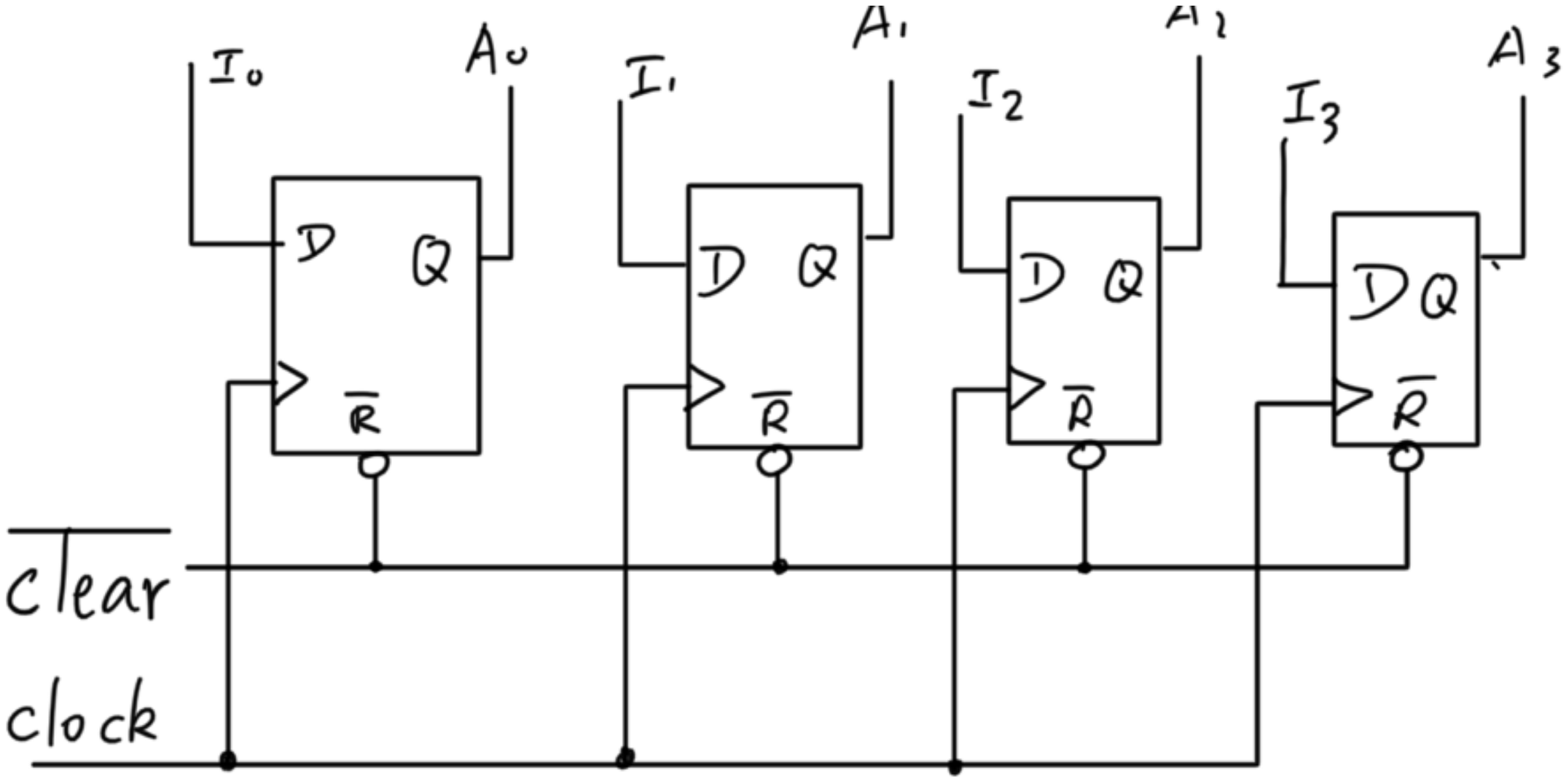
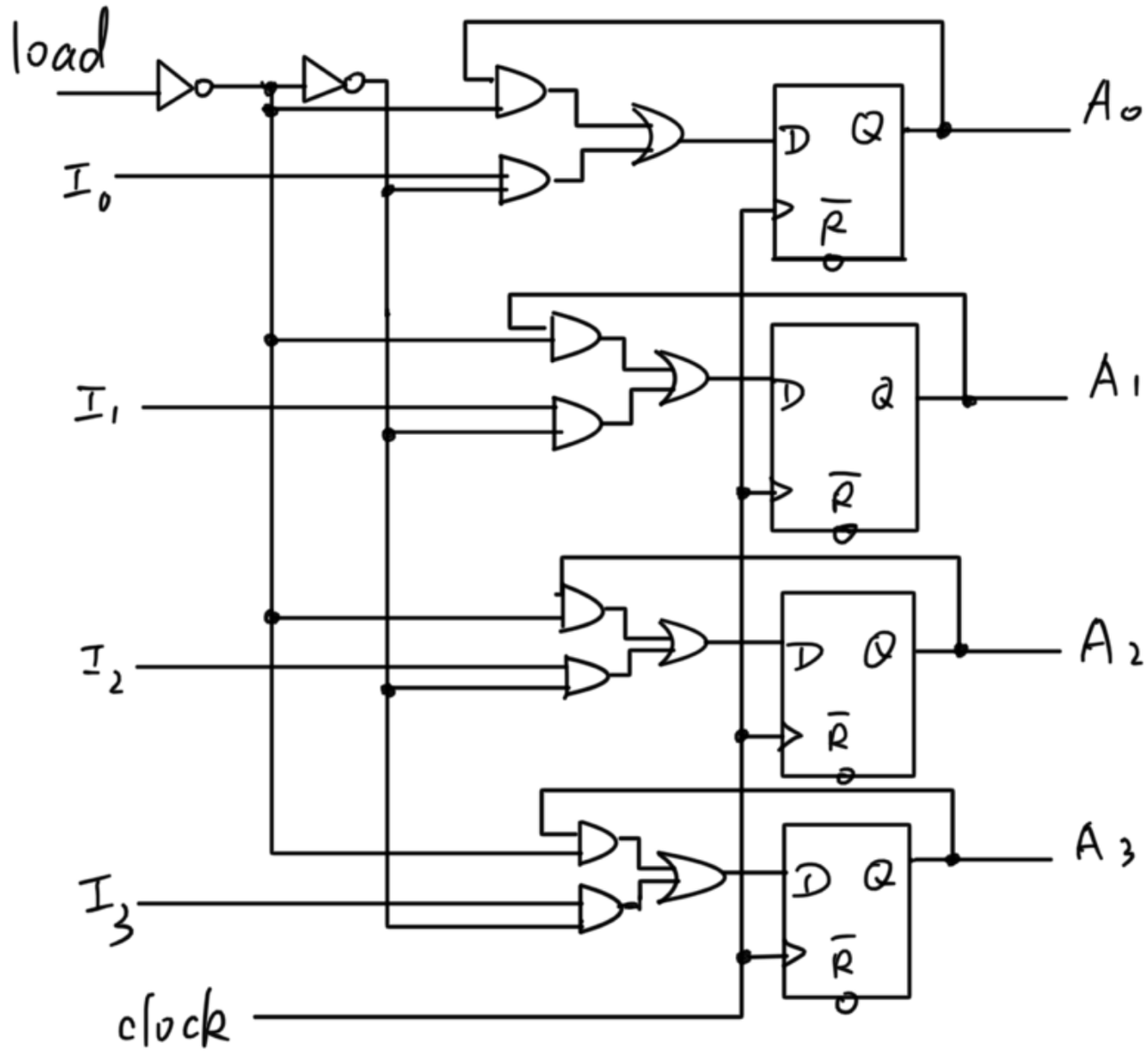
Digital Logic and Computer Organization

Sequential Logic —Applications

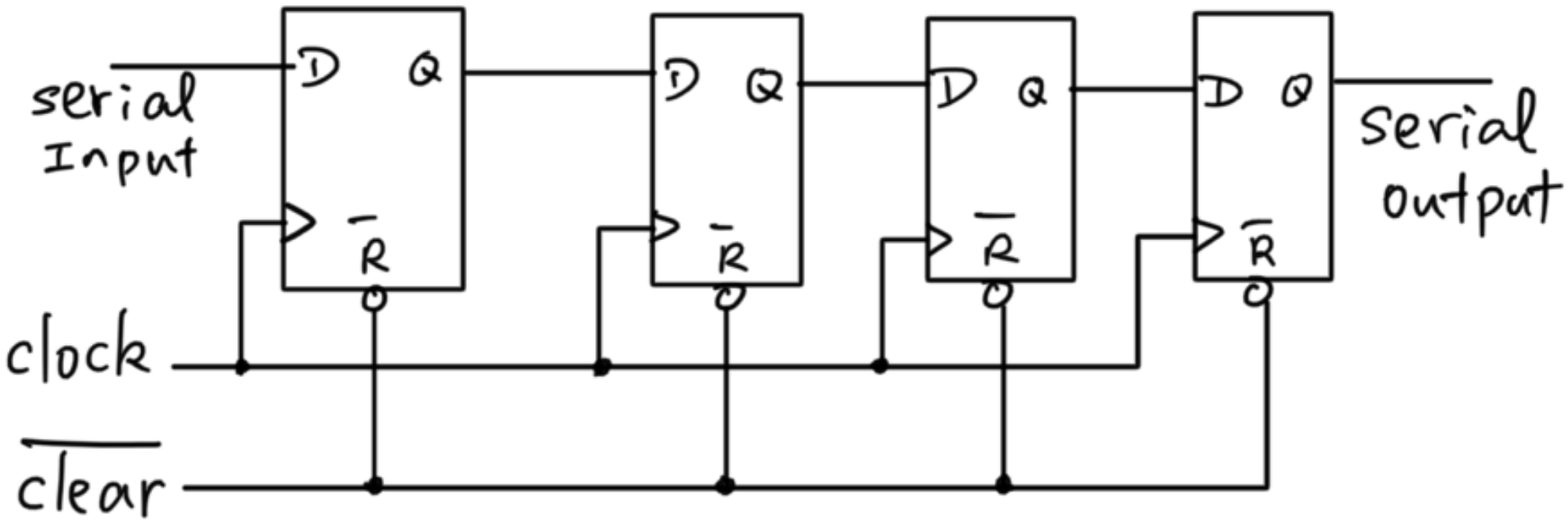
4-bit Register



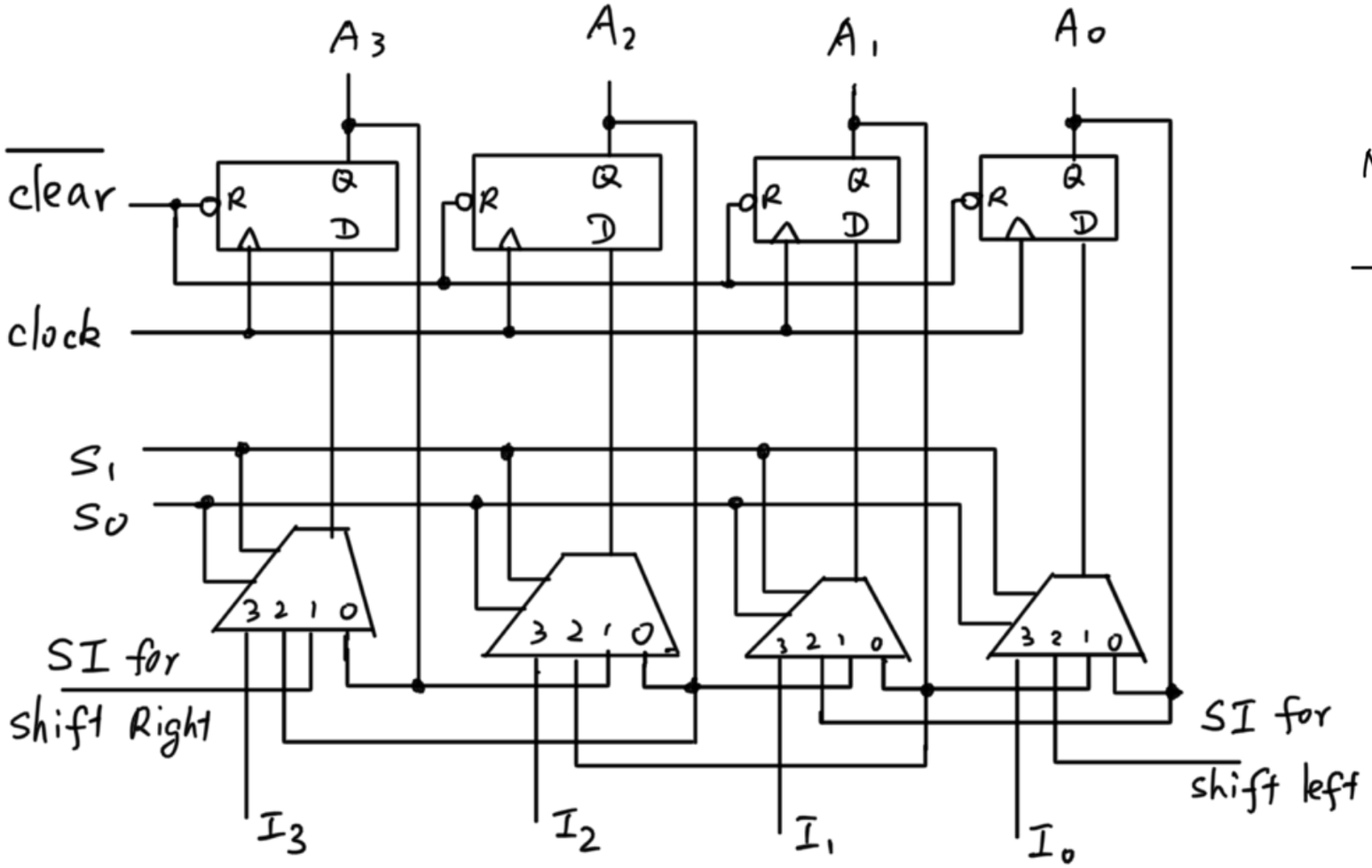
Register with Parallel Load



Shift Register

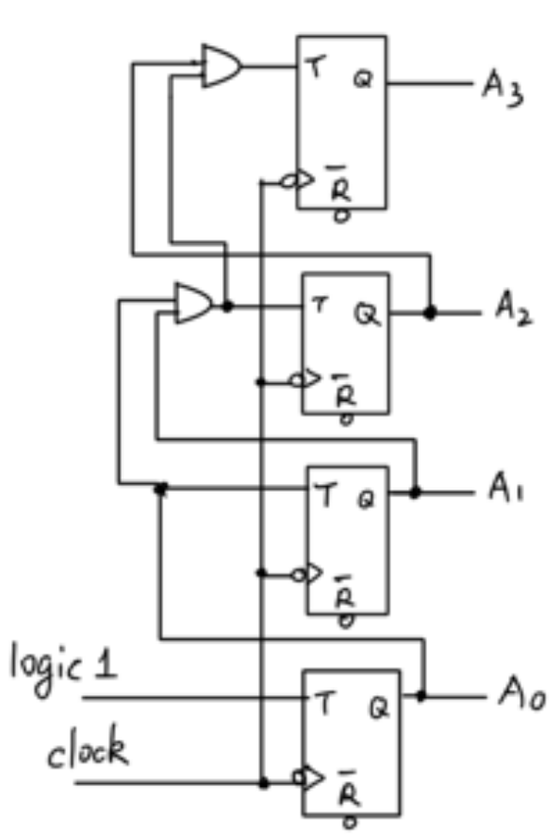


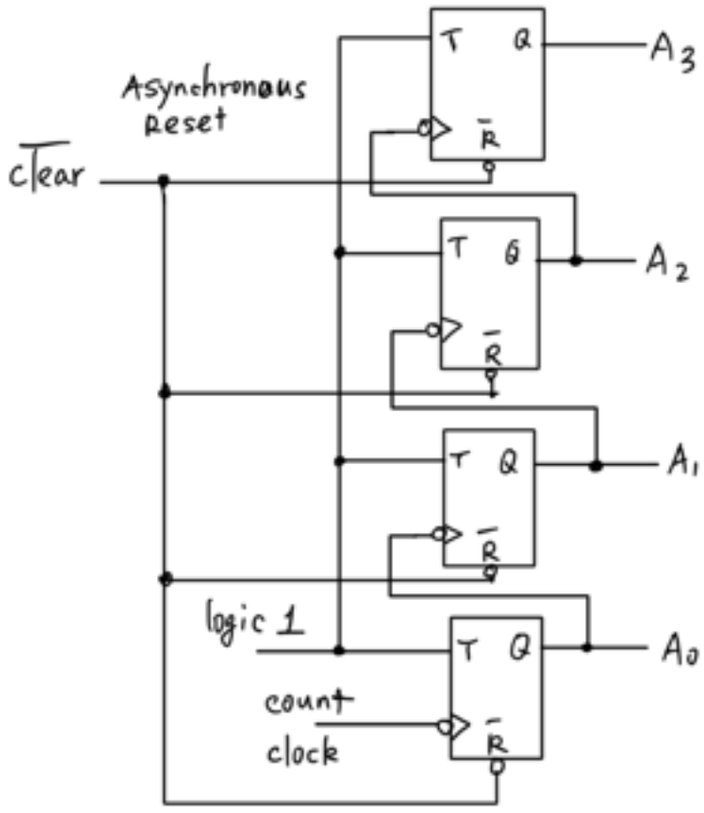
Universal Shift Register

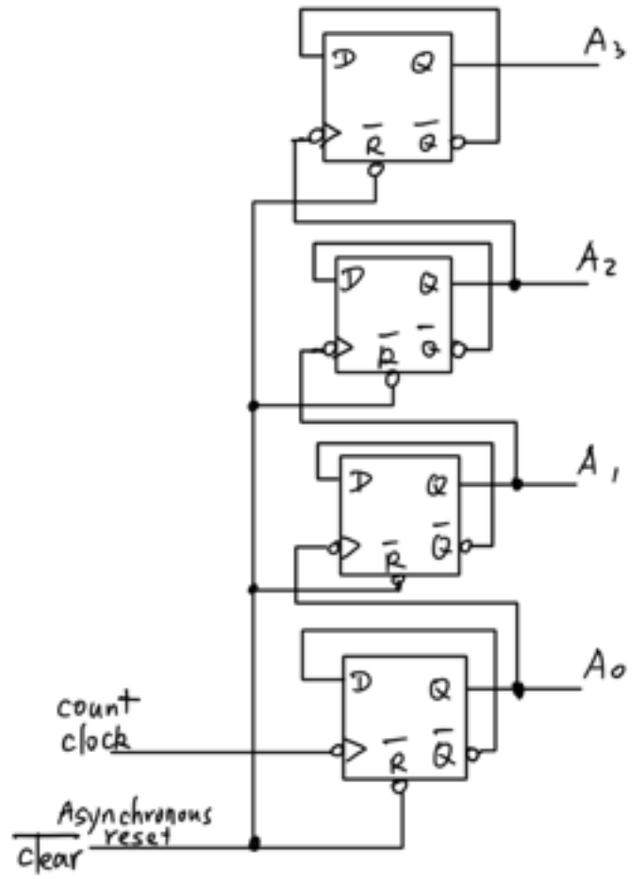


Mode Control S_1	S_0	Register operation
0	0	Hold
0	1	shift right
1	0	shift left
1	1	parallel load

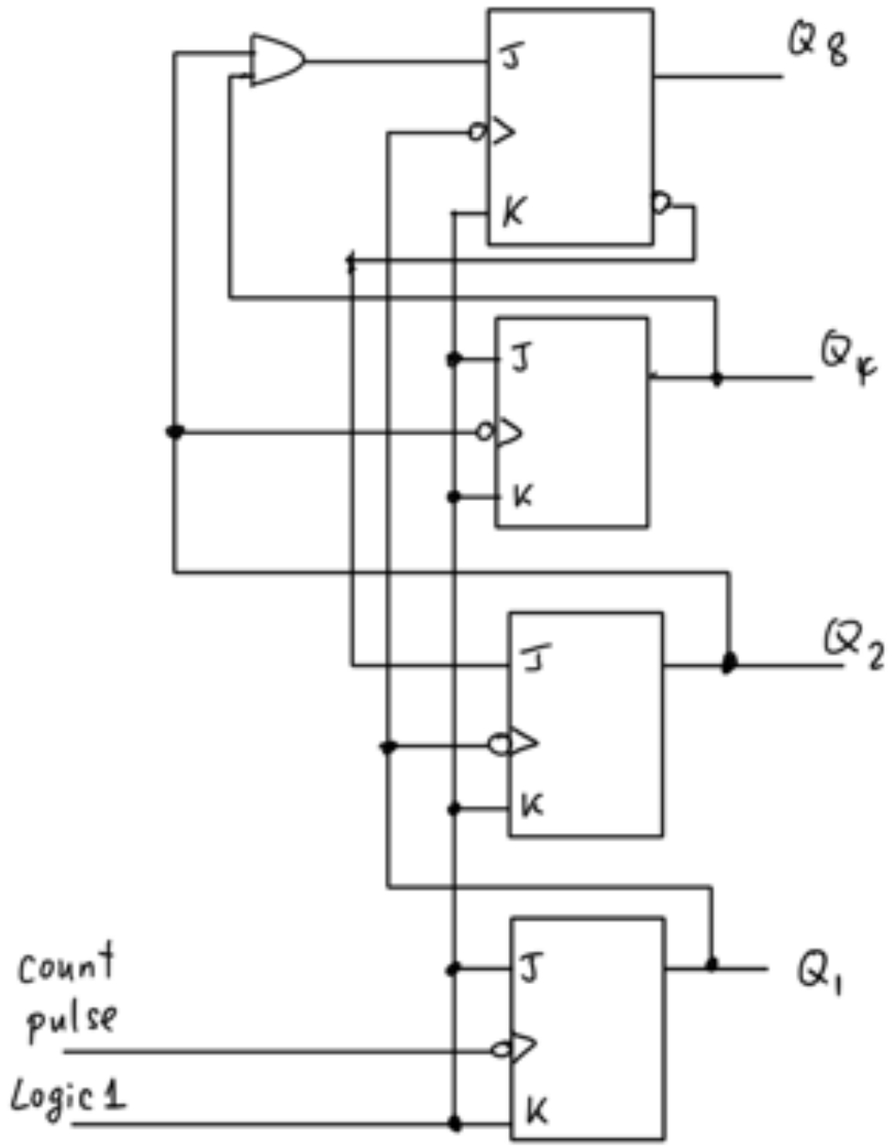
Counter

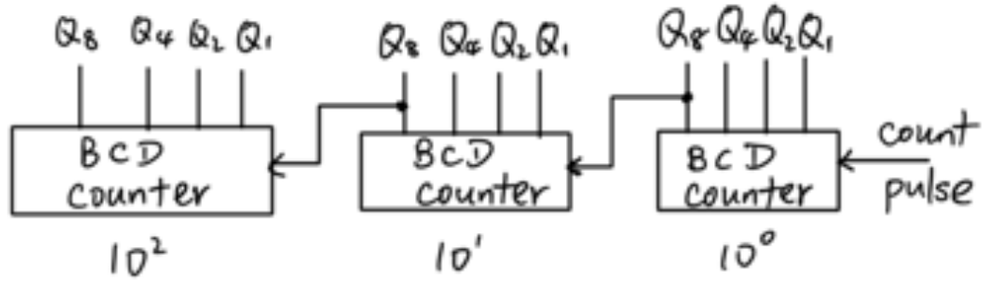

T
Q
$\overline{R}$
A_3
A_2
A_1
A_0
logic 1
clock


Asynchronous
Reset
clear
T
Q
$\overline{R}$
A_3
A_2
A_1
A_0
logic 1
count
clock

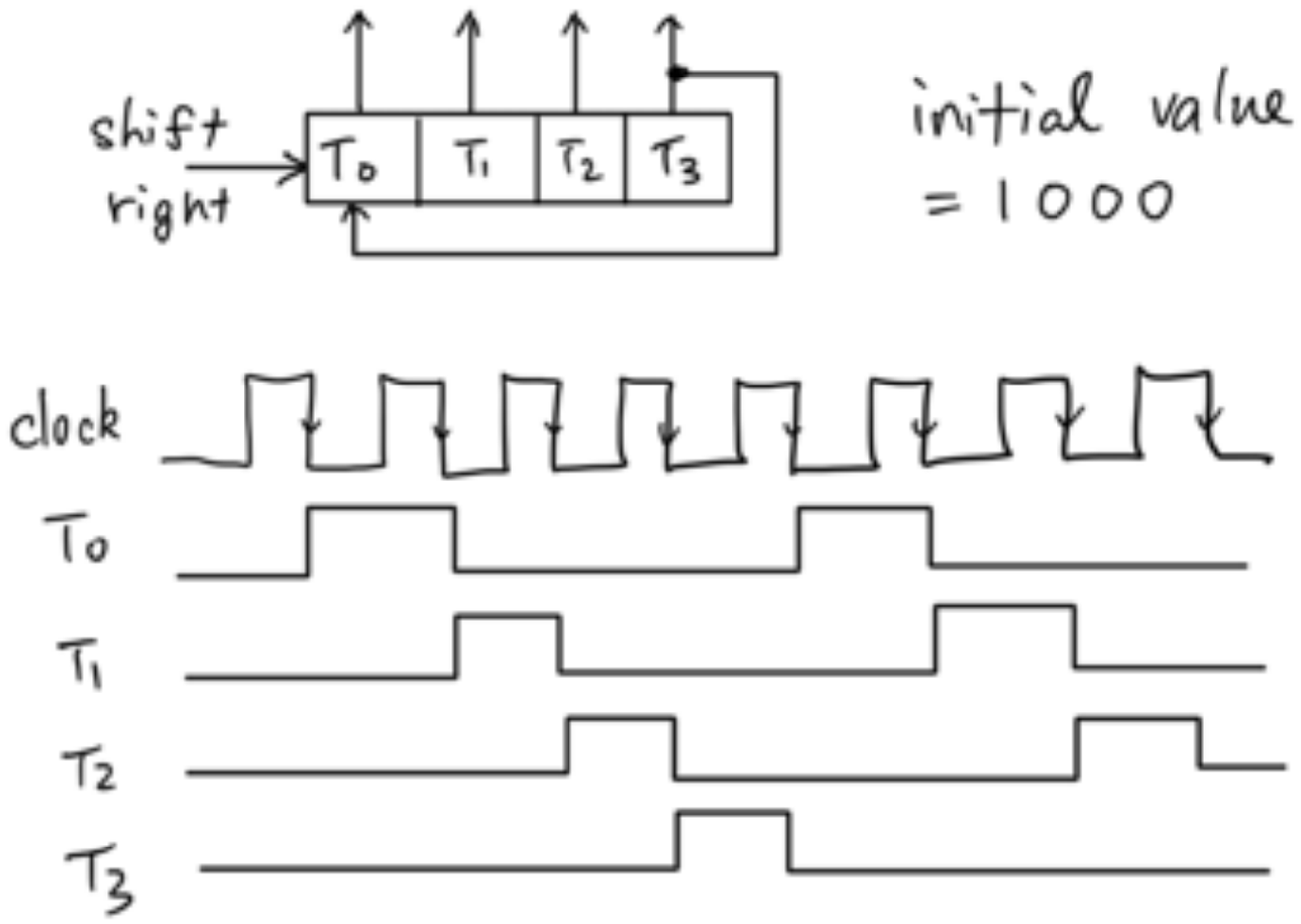

D
Q
$\overline{R}$
$\overline{Q}$
A_3
A_2
A_1
A_0
count
clock
Asynchronous
reset
clear

BCD Ripple Counter

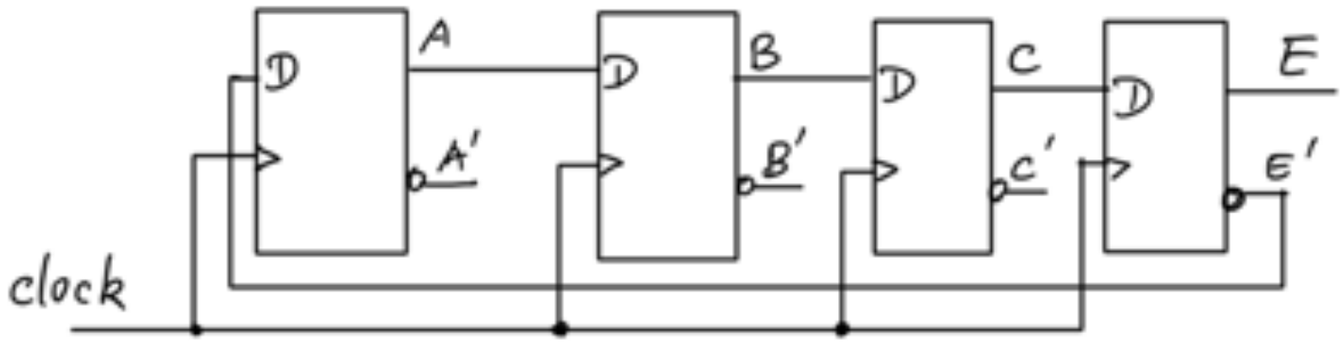




Ring Counter

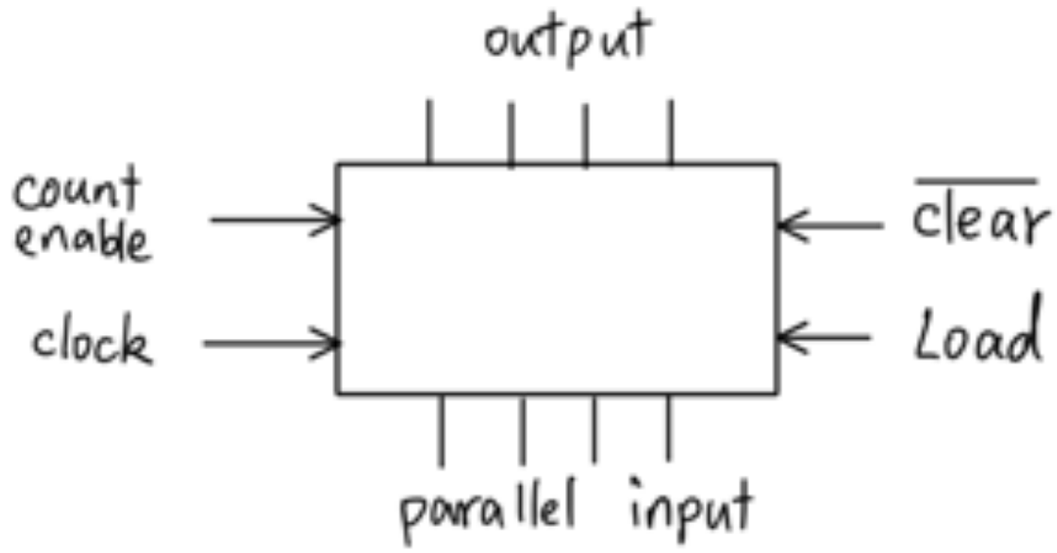


Johnson (Switch-tail) Ring Counter



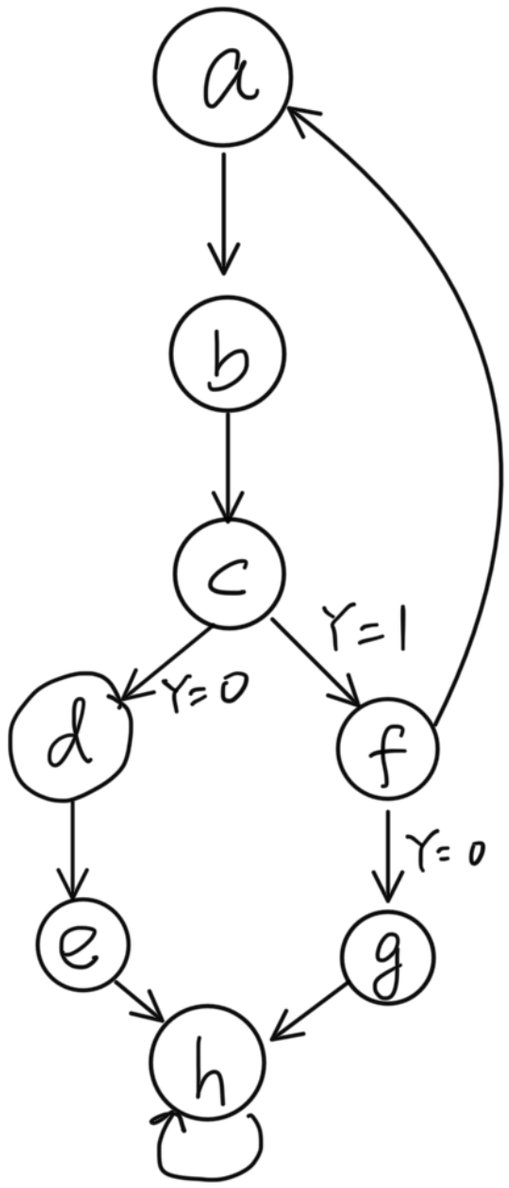
sequence	A	B	C	E	Output AND
1	0	0	0	0	$A'E'$
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	$A'B$
7	0	0	1	1	$B'C$
8	0	0	0	1	$C'E$

Binary Counter with Parallel Load



clear	clock	Load	count enable	Function
0	x	x	x	clear to 0
1	↑	1	x	load input
1	↑	0	1	count next
1	↑	0	0	no operation

Algorithmic State Machine (ASM) Chart



state	code
a	000
b	001
c	010
d	011
e	100
f	101
g	110
h	111

ASM Realization Using Counter

present state	Input Y	next state	$\overline{clear}$	Load	CE	parallel input
a	Y=0	b	1	0	1	x
	Y=1	b	1	0	1	x
b	Y=0	c	1	0	1	x
	Y=1	c	1	0	1	x
c	Y=0	d	1	0	1	x
	Y=1	f	1	1	x	101
d	Y=0	e	1	D	1	x
	Y=1	e	1	0	1	x
e	Y=0	h	1	1	x	111
	Y=1	h	1	1	x	111
f	Y=0	g	1	0	1	x
	Y=1	a	0	x	x	x
g	Y=0	h	1	0	1	x
	Y=1	h	1	0	1	x
h	Y=0	h	1	0	0	x
	Y=1	h	1	0	0	x

for state f Y=1, next state to be a
the input also can be:
$\overline{clear}$ =1, Load =1, CE = x, input =000