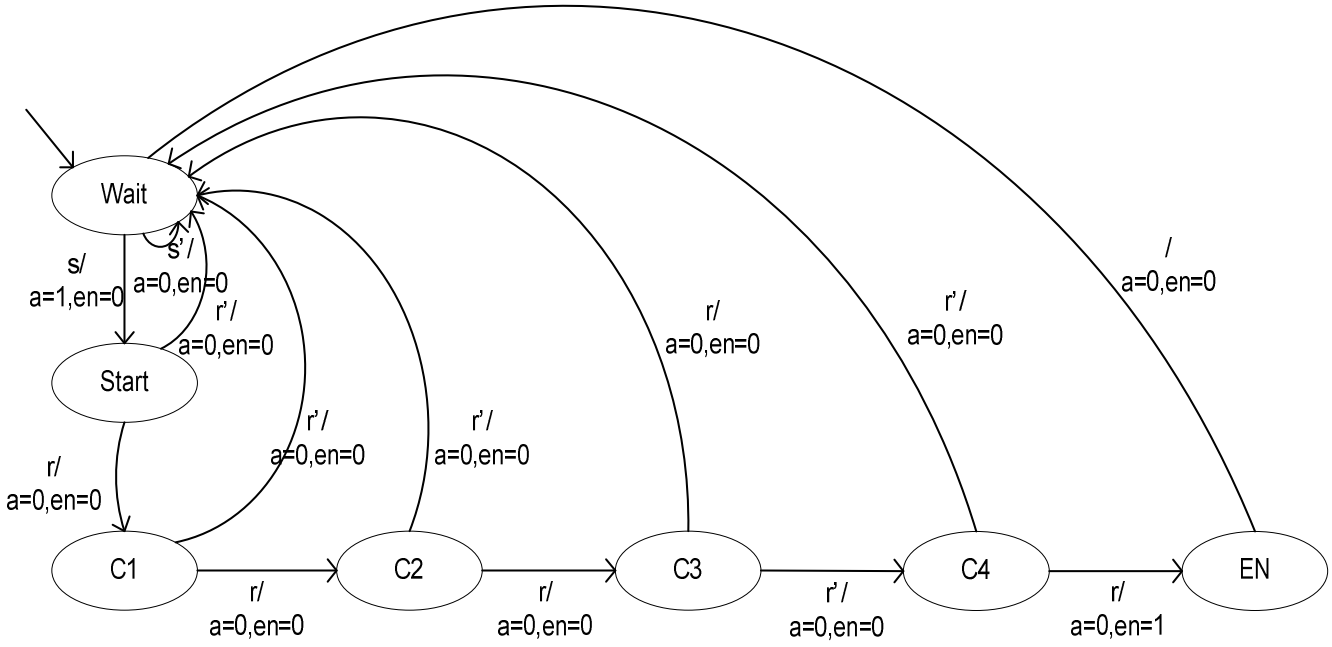
6.17

S1	(s3,s1) (s4,s2)					
S2	(s5,s1) (s6,s2)	(s5,s3) (s6,s4)				
S3	(s0,s1) (s0,s2)	(s0,s3) (s0,s4)	(s0,s5) (s0,s6)			
S4						
S5					(s0,s0) (s0,s0)	
S6	(s0,s1) (s0,s2)	(s0,s3) (s0,s4)	(s0,s5) (s0,s6)	(s0,s0) (s0,s0)		
	S0	S1	S2	S3	S4	S5

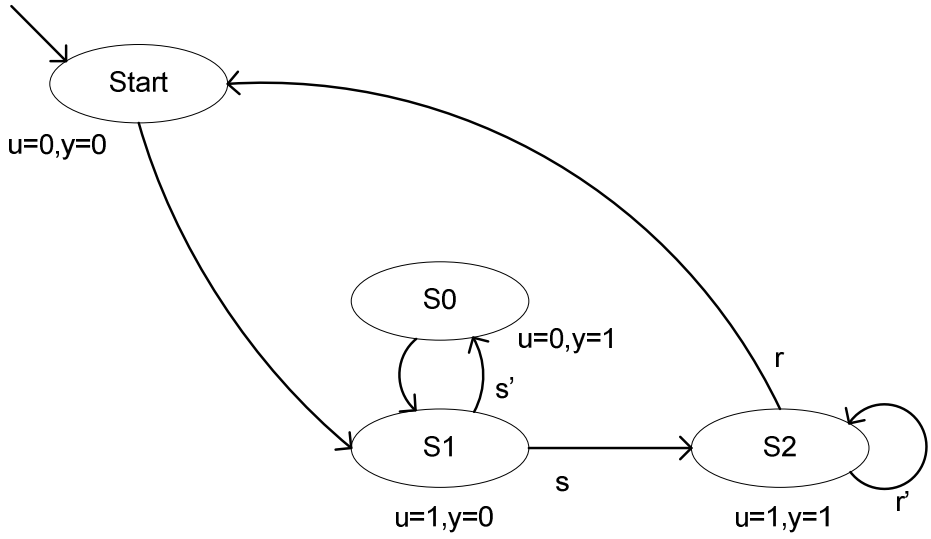
S1	(s3,s1) (s4,s2)					
S2	(s5,s1) (s6,s2)	(s5,s3) (s6,s4)				
S3	(s0,s1) (s0,s2)	(s0,s3) (s0,s4)	(s0,s5) (s0,s6)			
S4						
S5					(s0,s0) (s0,s0)	
S6	(s0,s1) (s0,s2)	(s0,s3) (s0,s4)	(s0,s5) (s0,s6)	(s0,s0) (s0,s0)		
	S0	S1	S2	S3	S4	S5

Therefore, state S3 = S6 and state S4 = S5.

6.23


Wait
Start
C1
C2
C3
C4
EN
s/
a=1,en=0
s' /
a=0,en=0
r' /
a=0,en=0
r/
a=0,en=0
r' /
a=0,en=0
r' /
a=0,en=0
r/
a=0,en=0
r' /
a=0,en=0
/
a=0,en=0
r/
a=0,en=0
r/
a=0,en=0
r/
a=0,en=0
r' /
a=0,en=0
r/
a=0,en=1

6.24


Start
u=0,y=0
S0
u=0,y=1
s'
S1
u=1,y=0
s
S2
u=1,y=1
r
r'

Minimal bit width encoding:
State encodings: S0: 000, S1: 001, S2: 010, S3: 011, S4: 100

Inputs					Outputs						
s2	s1	s0	B	S	n2	n1	n0	L	Dreg_clr	Dreg_ld	Dcnt_clr	Dcnt_cnt
0	0	0	x	x	0	0	1	0	1	0	0	0
0	0	1	0	x	0	0	1	0	0	0	1	0
0	0	1	1	x	0	1	0	0	0	0	1	0
0	1	0	x	x	0	1	1	1	0	0	0	0
0	1	1	x	0	0	1	1	0	0	0	0	1
0	1	1	x	1	1	0	0	0	0	0	0	1
1	0	0	x	x	0	0	1	0	0	1	0	0

n2 = s1s0S
n1 = s1’s0B + s1s0’ + s1s0S’
n0 = s1’s0’ + s1’s0B’ + s1s0’ + s1s0S’
L = s1s0’
Dreg_clr = s2’s1’s0’
Dreg_ld = s2
Dcnt_clr = s1’s0
Dcnt_cnt = s1s0
Logic size: 37 gate inputs
Delay: 2 gate delays

Output encoding:
State encodings: S0: 01000, S1: 00010, S2: 10000, S3: 00001, S4: 00100

Inputs							Outputs									
s4	s3	s2	s1	s0	B	S	n4	n3	n2	n1	n0	L	Dreg_clr	Dreg_ld	Dcnt_clr	Dcnt_cnt
0	1	0	0	0	x	x	0	0	0	1	0	0	1	0	0	0
0	0	0	1	0	0	x	0	0	0	1	0	0	0	0	1	0
0	0	0	1	0	1	x	1	0	0	0	0	0	0	0	1	0
1	0	0	0	0	x	x	0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	x	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	1	x	1	0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	x	x	0	0	0	1	0	0	0	1	0	0

n4 = s1’B
n3 = 0
n2 = s0S
n1 = s3 + s1x’ + s2
n0 = s4 + s0S’
L = s4
Dreg_clr = s3
Dreg_ld = s2
Dcnt_clr = s1
Dcnt_cnt = s0
Logic size: 13 gate inputs
Delay: 2 gate delays

6.26

Gi = aibi, Pi = ai xor bi

c1 = G0 + P0c0

c2 = G1 + P1G0 + P1P0c0

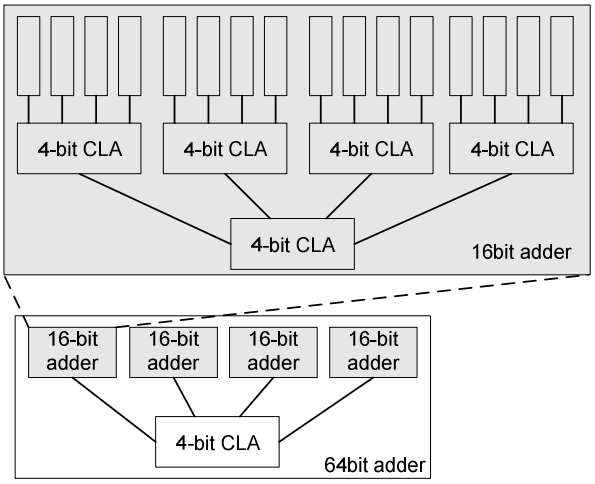
c3 = G2 + P2G1 + P2P1G0 + P2P1P0c0

c4 = G3 + P3G2 + P3P2G1 + P3P2P1G0 + P3P2P1P0c0

a = 11 = 1011, b = 7 = 0111

G0	1·1 = 1	P0	1 xor 1 = 0	c1	1 + 0·0 = 1
G1	1·1 = 1	P1	1 xor 1 = 0	c2	1 + 0·1 + 0·0·0 = 1
G2	0·1 = 0	P2	0 xor 1 = 1	c3	0 + 1·1 + 1·0·1 + 1·0·0·0= 1
G3	1·0 = 0	P3	1 xor 0 = 1	c4	0 + 1·0 + 1·1·1 + 1·1·0·1+ 1·1·0·0·0 = 1

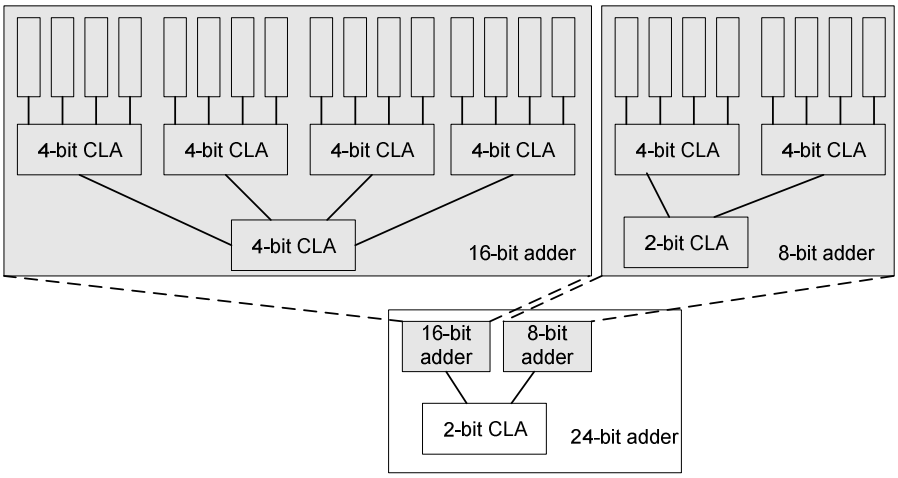
6.29



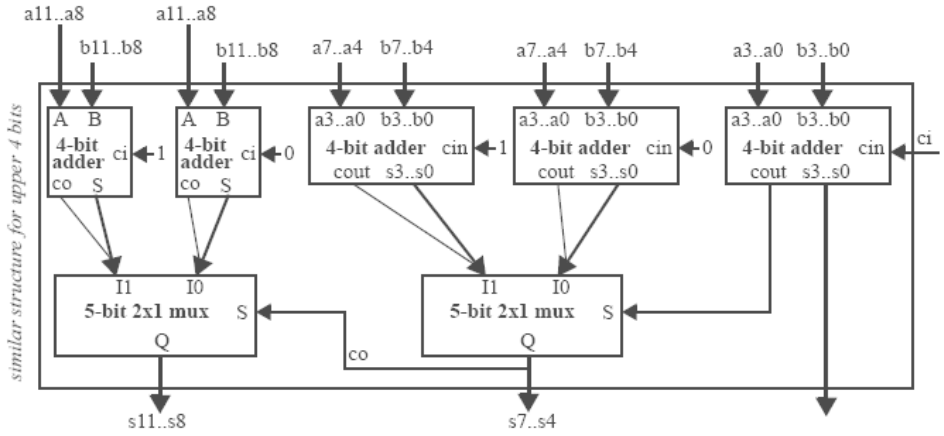
delay of 64-bit hierarchical carry-lookahead adder : 4 * 2 = 8, gate delay

delay of 64-bit carry-ripple adder : 64 * 2 = 128, 128 gate delay

6.30

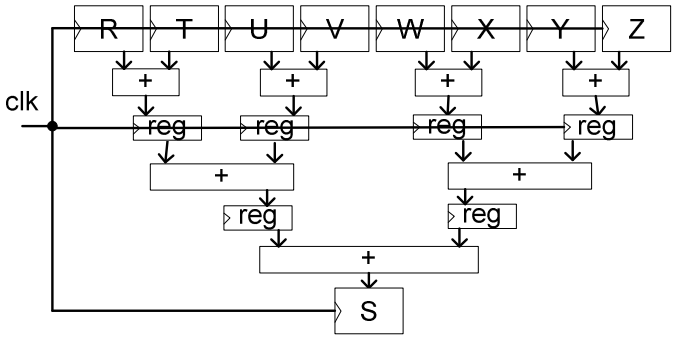

4-bit CLA
4-bit CLA
4-bit CLA
4-bit CLA
4-bit CLA
16-bit adder
4-bit CLA
4-bit CLA
2-bit CLA
8-bit adder
16-bit adder
8-bit adder
2-bit CLA
24-bit adder

6.31

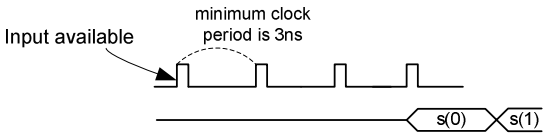

a11..a8
a11..a8
b11..b8
b11..b8
a7..a4
b7..b4
a7..a4
b7..b4
a3..a0
b3..b0
similar structure for upper 4 bits
A B
4-bit adder
ci
co S
1
0
a3..a0 b3..b0
4-bit adder cin
cout s3..s0
ci
I1 I0
5-bit 2x1 mux
S
Q
co
s11..s8
s7..s4

6.32

(a)


R
T
U
V
W
X
Y
Z
clk
+
reg
S

(b) assume that delay adder is 3ns.



6.33

in case of 6.94, the minimum clock period must be 9ns.

in case of 6.32, the minimum clock period must be 3ns.

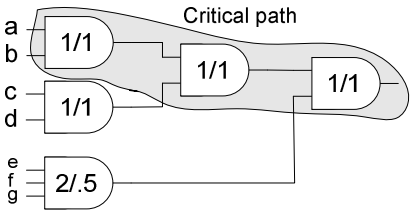
Therefore, pipelined version is three times faster.

6.34

latency : 9ns

throughput : 1 sample every 3 ns

6.44



6.45