

Exam solutions ASIC (TSTE81) 990825

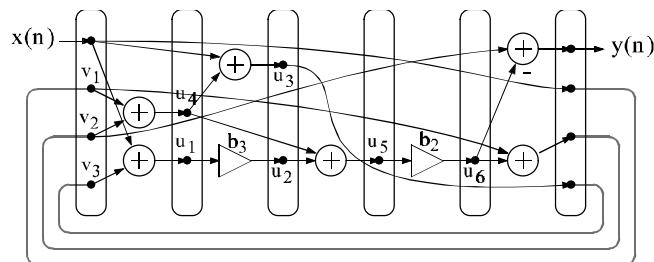
1. a) Ja. Se problem 7.4 i boken.

b) Ja. Talet noll kan bara representeras på ett sätt i SDC. Samma gäller även för maximalt positiva och maximalt negativa talet i talområdet.

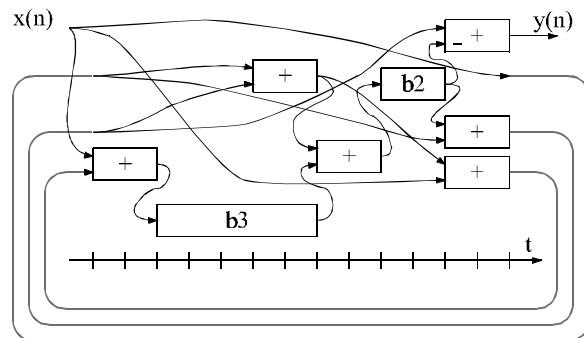
c) Crossbar.

d) Nej. ROM:en innehåller summor av koefficienter, och dessa summor kan då kräva längre ordlängd än någon enskild koefficient.

e) $T_{\text{sample}} = T_{\text{min}}$



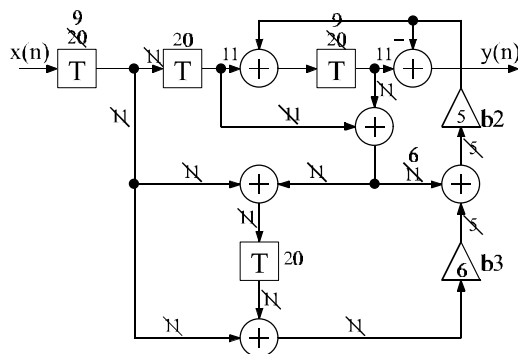
2. a)



b)

3. a) $T_{\text{min}} = \max \{ (T_{\text{add}} + T_{\text{add}} + T_{b2} + T_{\text{add}})/1, (T_{\text{add}} + T_{\text{add}} + T_{\text{add}} + T_{b3} + T_{\text{add}} + T_{b2} + T_{\text{add}})/2 \} = \max \{ (0+0+5+0)/1, (0+0+0+6+0+5+0)/2 \} = \max(5, 5.5) = 5.5$ tidsenheter.

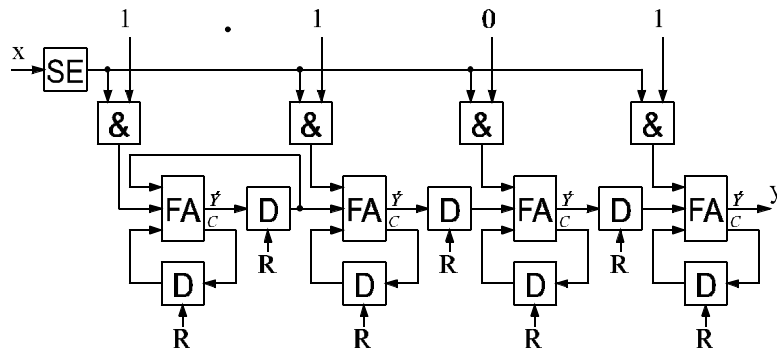
b) Ersätt alla fördröjningselement med 20 vippor, och lägg till 20 vippor på ingången. Propagera ut dessa så alla operationer får tilldelat tid för exekvering.



Exam solutions ASIC (TSTE81) 990825

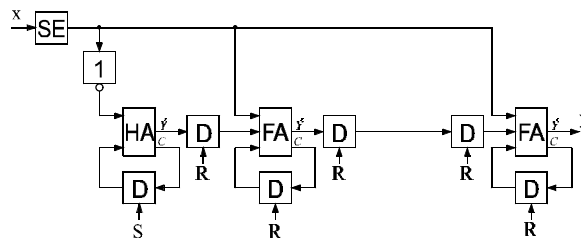
- c) Latency = tid för en bit att propageras från in till utgång. Innanshimming-delay adderats är latency = 0. 20 vippor läggs till (= 1 samplperiod). Av dessa används endast 14 vippor. De återstående 6 vipporna kan tas bort. Därför är latency $14/20 = 0,7$ samplperioder.

4. a) $-0.375 = -0.011 = 1.101_{(2C)}$.



Reset (R) nollställer vipporna vid start av beräkningen. SE teckenförlänger indata när teckenbiten matats in i kretsen, men utdata inte är färdigt.

- b) Den första bit-slicen kan bytas mot en subtraktion (som syns nedan). Om inte så måste hela första slicen bevaras (helt godkänt det också).



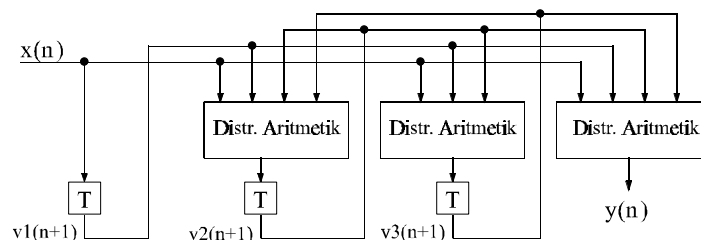
5. a) Dom summor av produkter som måste beräknas är de indata som behövs av T-elementen samt utsignalen. Alltså beräknas

$$v1(n+1) = x(n)$$

$$v2(n+1) = v1(n) + b2(v2(n) + v1(n) + b3(v3(n) + x(n)))$$

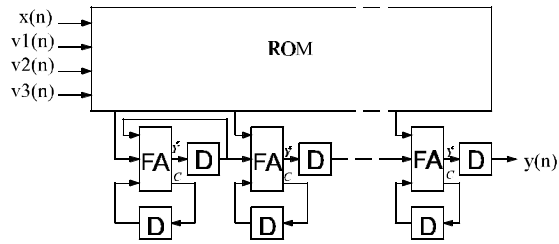
$$v3(n+1) = x(n) + v1(n) + v2(n)$$

$$y(n) = v2(n) - b2(v1(n) + v2(n) + b3(v3(n) + x(n)))$$



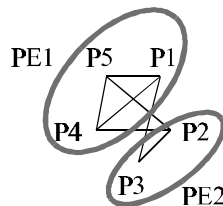
- b) Både $y(n)$ och $v2(n)$ har lika många ingångar. Betrakta $y(n)$.

Exam solutions ASIC (TSTE81) 990825



Uttrycket som skall beräknas är: $y(n) = -b_3b_2x(n) - b_2v_1(n) + (1-b_2)v_2(n) - b_2b_3v_3(n)$

x v1 v2 v3	ROM
0 0 0 0	0
0 0 0 1	-b ₂ b ₃
0 0 1 0	1-b ₂
0 0 1 1	1-b ₂ -b ₂ b ₃
0 1 0 0	-b ₂
0 1 0 1	-b ₂ -b ₂ b ₃
0 1 1 0	1-2b ₂
0 1 1 1	1-2b ₂ -b ₂ b ₃
1 0 0 0	-b ₂ b ₃
1 0 0 1	-2b ₂ b ₃
1 0 1 0	1-b ₂ -b ₂ b ₃
1 0 1 1	1-b ₂ -2b ₂ b ₃
1 1 0 0	-b ₂ -b ₂ b ₃
1 1 0 1	-b ₂ -2b ₂ b ₃
1 1 1 0	1-b ₂ b ₃ -2b ₂
1 1 1 1	1-2b ₂ b ₃ -2b ₂



6. a)

- b) Om man antar att både läsning och skrivning sker parallellt så sker 4 simultiga överföringar mellan processorelement och minnen. P1->P4, Minne->P4, P2->Minne, Minne->P3. Om överföringarna från processorelement sker skilt från överföring till processorelement (tidsmässigt) så finns det maximalt 3 simultiga överföringar (->P4, ->P4, ->P3). Alltså beroende på antagandet om kommunikationen så behövs antingen minst 3 eller minst 4 bussar för att implementera schemuleringen.