

1.
 - a) A constructive algorithm constructs the solution from scratch, while an iterative improves a previous gives solution.
 - b) Systolic: synchronous. Wave front: asynchronous
 - c) Carr look-ahead, carry-select, carry-increment, conditional-sum.
 - d) Yes, 0 and the maximum and minimum number in the number range.
 - e) Verification: exhaustive testing. Validation: non-exhaustive testing.

2. Följande första ordningens överföringsfunktion för ett filter är given:
The following first-order filter transfer function is given:

$$H(z) = \frac{z}{z-0.5} = \frac{1}{1-0.5z^{-1}}$$

- a) Clustered lookahead:

$$H(z) = \frac{1}{1-0.5z^{-1}} \left(\frac{1+az^{-1}+bz^{-2}}{1+az^{-1}+bz^{-2}} \right) = \frac{1+az^{-1}+bz^{-2}}{1+(a-0.5)z^{-1}+(b-0.5a)z^{-2}-0.5bz^{-3}}$$

$$a-0.5 = 0 \Rightarrow a = 0.5$$

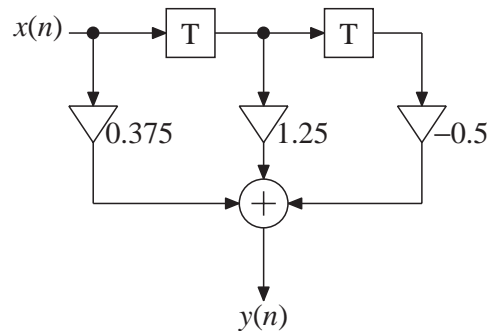
$$b-0.5a = 0 \Rightarrow b = 0.5a = 0.25$$

$$H(z) = \frac{1+0.5z^{-1}+0.25z^{-2}}{1-0.125z^{-3}} = \frac{z(z^2+0.5z+0.25)}{z^3-0.125}$$

Scattered lookahead

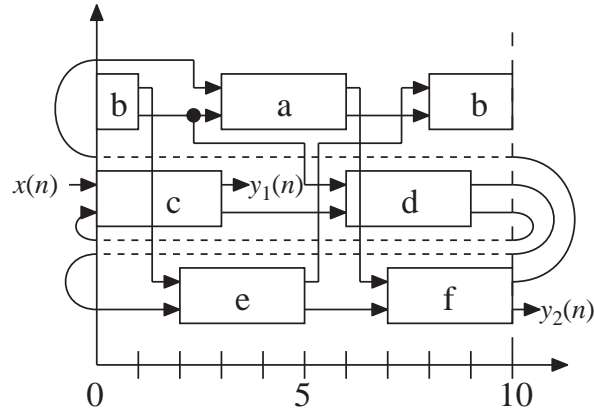
- b) Yes! All introduced poles inside the unit circle.

3. FIR-filtret nedan skall implementeras genom att göra en isomorfisk mappning till bit-seriella beräkningselement. Dataordlängden är 15 bitar. Använd modell 0, dvs inga extra pipeliningregister. *The FIR filter below is to be implemented using isomorphic mapping to bit-serial processing elements. The data wordlength is 15 bits. Assume model 0, i.e., no extra pipelining registers.*



- Rita en fullt specificerad signalflödesgraf. *Draw a fully specified signal flow graph.* (2)
- Introducera shimming delay. *Introduce shimming delays.* (6)
- Beskriv arkitekturen för en implementering med distribuerad aritmetik. Använd byggblock som skiftackumulatörer, skiftregister, ROM, etc. *Describe the architecture for an implementation using distributed arithmetic. Use building blocks such as shift accumulators, shift registers, ROM, etc.* (4)
- Bestäm ROM-innehållet om filtret implementeras med en distribuerad aritmetikenhet. Använd en lämplig binär representation. *Determine the ROM contents when the filter is implemented with one distributed arithmetic unit. Use a suitable binary representation.* (4)

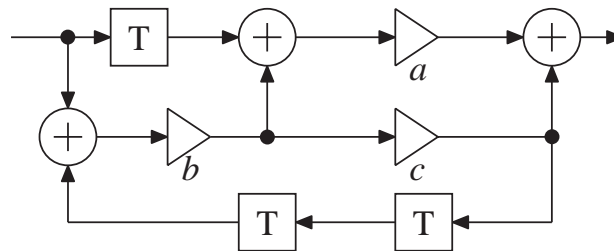
4. Schemat nedan är givet. Processorelementen är homogena och icke-preemptiva. Namngivningen är till för referensändamål. *The schedule below is given. The processing elements are homogeneous and non-preemptive. The naming is used for referencing.*



- a) Utför resursallokering och tilldelning för processorelementen med clique partitionering. *Perform resource allocation and assignment of the processing elements using clique partitioning.* (6)
- b) $N_{PE, min} = \left\lceil \frac{3 \cdot 6}{10} \right\rceil = 2 \text{ PEs}$
- c) Utför resursallokering och tilldelning för minnesvariablerna med vänsterkantsalgoritmen (left-edge algoritmen) och redovisa alla mellansteg. Antag att skrivning och läsning **ej** kan ske i samma tidslucka för en minnescell, dvs minnesvariablerna får inte ligga "kant-i-kant". In- och utsignaler behöver ej tas hänsyn till. *Perform resource assignment and allocation of the memory variables using the left-edge algorithm and show all intermediate steps. Assume that it is **not** possible to read from and write to a memory cell in the same time slot, i.e., memory variables can not be placed "edge-to-edge". Input and output signals should not be considered.* (6)

5. Filtret nedan skall implementeras med en arkitektur som använder delat minne. Beräkningselementen är non-preemptive och icke-homogena. Latencyn för tre tidsenheter för multiplikation a , fyra tidsenheter för multiplikation b och fem tidsenheter för multiplikation c . För additioner är latencyn en tidsenhet.

The filter below is to be implemented using a shared memory architecture. The processing elements are non-preemptive and non-homogenous. The latency is three time units for multiplication a , four time units for multiplication b , and five time units for multiplication c . Each addition has a latency of one time unit.



a)
$$T_{min} = \frac{T_b + T_c + T_{add}}{2} = \frac{4 + 5 + 1}{2} = 5 \text{ time units}$$

b) $T_{cp} = 11 \text{ time units}$

- c) Rita signalflödesgrafan i precedensform.

Draw the signal-flow graph in precedence form. (6)

- d) Schemulera algoritmen så att $T_{sample} = T_{min}$.

Schedule the algorithm so that $T_{sample} = T_{min}$. (8)

- e) Använd pipelining för att reducera tiden för den kritiska vägen till T_{min} . Använd minimalt med fördröjningsselement.

Use pipelining to reduce the time of the critical path to T_{min} . Use a minimum number of delay elements. (4)