

# Written Examination TSTE87

2015-08-19

1. (a) What is the difference between an iterative and a block processing DSP algorithm? Give an example of each! (2 p)

**Solution:**

- (b) Define latency and execution time. (2 p)  
(c) Name exactly two features that are typically found in general purpose DSP processors, but not in CPUs/microcontrollers. (2 p)

**Solution:**

- (d) Explain how a minimum signed-digit representation and the canonic signed-digit representation differs from the general signed-digit representation. (2 p)

**Solution:**

- (e) Name and describe one way to accelerate a carry-propagation adder. (2 p)
2. Consider the memory variable life time diagram in Fig. 1, consisting of variables a to j, which should be mapped to a single memory. The memory can read from and write to the same address in the same cycle.

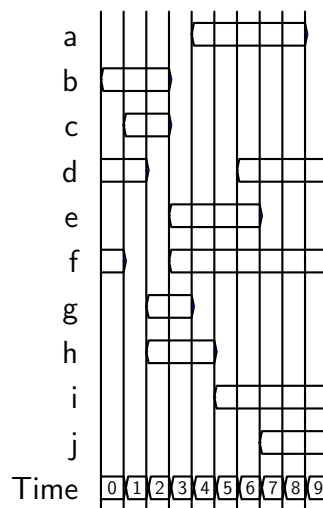
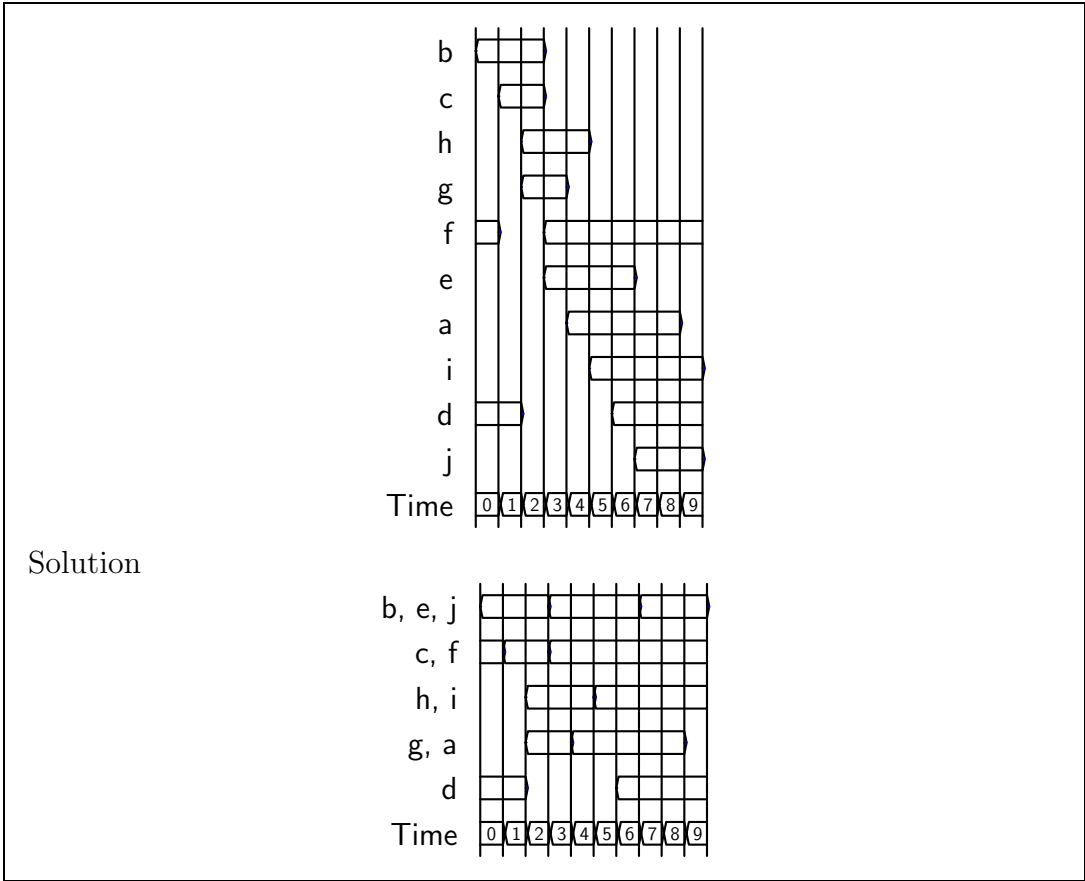


Figure 1: Life time diagram for Problem 2.

- (a) Assign the variables to a minimal number of memory cells. (6 p)

**Solution:** Use the left edge algorithm for each memory.  
Sorted:



(b) How many read and write ports are required for the memory, respectively?

(2 p)

3. Consider the implementation of the seventh-order lattice wave digital filter shown in Fig. 2, which should operate at a sample rate of 80 MHz. This should be implemented using a shared memory architecture and one of two options for processing elements. In option I separate adders/subtractors and multipliers are used, see Fig. 3 for an SFG of the symmetric two-port adaptor used. In option II, a processing element forming a complete symmetric two-port adaptor is used. The processing elements have an execution time of 4 ns, i.e., they can operate at 250 MHz.

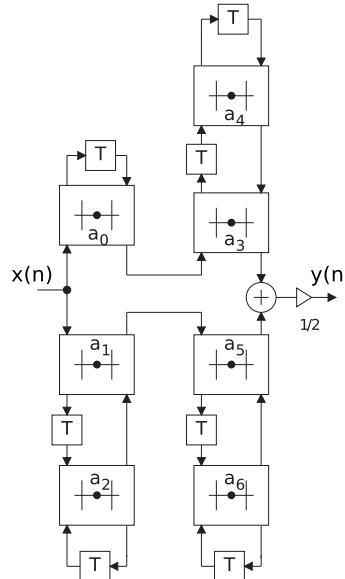


Figure 2: Seventh-order lattice wave digital filter for Problem 3.

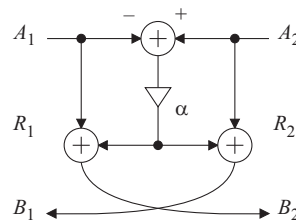


Figure 3: SFG of symmetric two-port adaptor for use in Problem 3.

- (a) Determine the number of processing elements required for options I and II, respectively. The final addition and scaling can in option II be replaced with an adaptor operation. Ignore the scaling for option I. (4 p)

**Solution: Option I**

Seven multiplications per sample leading to  $7 \times 80 = 560$  Mmult/s.

$$\left\lceil \frac{7 \times 80 \times 10^6}{250 \times 10^6} \right\rceil = 3 \text{ multiplier PEs}$$

22 add/sub per sample leading to

$$\left\lceil \frac{22 \times 80 \times 10^6}{250 \times 10^6} \right\rceil = 8 \text{ add/sub PEs}$$

**Option II**

Eight adaptor operations per sample leading to

$$\left\lceil \frac{8 \times 80 \times 10^6}{250 \times 10^6} \right\rceil = 3 \text{ adaptor PEs}$$

- (b) Determine the memory bandwidth (number of reads and writes per second, respectively) for options I and II, respectively. Assume that the input and output do not have to be stored in the memory. (4 p)

**Solution: Option I**

Each multiplication leads to one read and one write (coefficients are dealt with separately). Each add/sub leads to two reads and one write.

$7 + 22 \times 2 - 2 = 49$  reads per sample ( $-2$  as the input goes to two adders)

$$49 \times 80 \times 10^6 = 3.92 \text{ Gread/s}$$

$7 + 22 - 1 = 28$  writes per sample ( $-1$  as the output does not have to be written)

$$28 \times 80 \times 10^6 = 2.24 \text{ Gwrite/s}$$

**Option II** Each adaptor leads to two reads and two writes

$8 \times 2 - 2 = 14$  reads per sample ( $-2$  as the input goes to two adaptors)

$$14 \times 80 \times 10^6 = 1.12 \text{ Gread/s}$$

$8 \times 2 - 2 = 14$  writes per sample ( $-2$  as the output does not have to be written plus that the second output of adaptor used for adding and scaling is not use)

$$14 \times 80 \times 10^6 = 1.12 \text{ Gwrite/s}$$

- (c) Show that the final addition and scaling can be realized using a symmetric two-port adaptor as shown in Fig. 3. (2 p)

**Solution:** The adaptor computes

$$B_1 = -\alpha A_1 + (1 + \alpha)A_2$$

$$B_2 = (1 - \alpha)A_1 + \alpha A_2$$

By selecting  $\alpha = \frac{1}{2}$  ( $\alpha = -\frac{1}{2}$ ) the expected result  $\frac{A_1+A_2}{2}$  is obtained at  $B_2$  ( $B_1$ ).

4. The digital filter algorithm in Fig. 4 shall be implemented on a shared-memory architecture with homogeneous non-preemptive processing elements. There are two types of processing elements: multipliers and adders. The latency of a multiplier is three clock cycles and the latency of an adder is two clock cycles. Both types of processing elements have an execution time of one clock cycle.

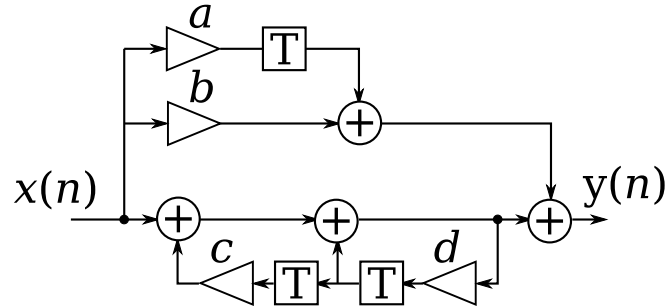


Figure 4: Digital filter for use in Problems 4 and 5.

- (a) Determine the minimal sample period,  $T_{\min}$ . (2 p)

**Solution:**

$$T_{\min} = \max \left\{ \frac{T_{\text{add}} + T_{\text{mult}}}{1}, \frac{2T_{\text{add}} + 2T_{\text{mult}}}{2} \right\} = T_{\text{add}} + T_{\text{mult}} = 5 \text{ clock cycles}$$

- (b) Determine the critical path,  $T_{cp}$ . (2 p)

**Solution:**

$$T_{cp} = 2T_{\text{add}} + 2T_{\text{mult}} = 10 \text{ clock cycles}$$

- (c) Determine the precedence graph. (6 p)
- (d) Schedule the algorithm for  $T_s = T_{\min}$  and a reasonably efficient resource utilization. Make sure that you indicate all the connections in the schedule. (8 p)
- (e) How many processing elements are needed of each type? What are the theoretical limit? Comments? (2 p)

**Solution:**

$$N_{\text{PE}} = \left\lceil \frac{\sum T_{\text{exe}}}{T_s} \right\rceil$$

$$N_{\text{mult}} = \left\lceil \frac{4 \times 1}{5} \right\rceil = 1 \text{ multiplier}$$

$$N_{\text{add}} = \left\lceil \frac{4 \times 1}{5} \right\rceil = 1 \text{ adder}$$

(f) Now, assume that the algorithm should be implemented using MAC operations,  $x \times y + z$ . Apply transformations to the algorithm to result in only MAC operations (four MAC operations are enough). (4 p)

(g) With a latency of four clock cycles for the MAC operation, determine the minimal sample period,  $T_{\min}$  (2 p)

**Solution:**

$$T_{\min} = \max \left\{ \frac{T_{\text{MAC}}}{1}, \frac{2T_{\text{MAC}}}{2} \right\} = T_{\text{MAC}} = 4 \text{ clock cycles}$$

(h) Determine the critical path,  $T_{cp}$ , for the MAC-based algorithm. (2 p)

**Solution:**

$$T_{cp} = 3T_{\text{MAC}} = 12 \text{ clock cycles}$$

5. The filter in Fig. 4 is now to be implemented using distributed arithmetic. The filter coefficients are  $a = \frac{45}{64}$ ,  $b = -\frac{13}{8}$ ,  $c = -\frac{7}{16}$ , and  $d = \frac{3}{4}$ .

(a) Perform retiming to reduce the number of distributed arithmetic units (two are enough). (2 p)

(b) Determine the state-space representation of the retimed algorithm. (2 p)

**Solution:**

$$\begin{bmatrix} v_1(n+1) \\ v_2(n+1) \\ v_3(n+1) \\ y(n) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & d & cd & d \\ 0 & 1 & 0 & 0 \\ a & 1 & c & 1+b \end{bmatrix} \begin{bmatrix} v_1(n) \\ v_2(n) \\ v_3(n) \\ x(n) \end{bmatrix}$$

(c) Draw the resulting architecture using building blocks such as shift-registers, shift-accumulators, and ROMs. (4 p)

(d) Determine the ROM contents in a named binary representation. (4 p)

**Solution:**

$v_2$	$v_3$	$x$	Value	Two's complement
0	0	0	0	00.000000
0	0	1	$\frac{3}{4}$	00.110000
0	1	0	$-\frac{21}{64}$	11.101011
0	1	1	$\frac{27}{64}$	00.011011
1	0	0	$\frac{3}{4}$	00.110000
1	0	1	$\frac{3}{4}$	01.100000
1	1	0	$\frac{27}{64}$	00.011011
1	1	1	$\frac{75}{64}$	01.001011

$v_1$	$v_2$	$v_3$	$x$	Value	Two's complement
0	0	0	0	0	00.000000
0	0	0	1	$-\frac{5}{8}$	11.011000
0	0	1	0	$-\frac{7}{16}$	11.100100
0	0	1	1	$-\frac{17}{16}$	10.111100
0	1	0	0	1	01.000000
0	1	0	1	$\frac{3}{8}$	00.011000
0	1	1	0	$\frac{9}{16}$	00.100100
0	1	1	1	$-\frac{1}{16}$	11.111100
1	0	0	0	$\frac{45}{64}$	00.101101
1	0	0	1	$\frac{5}{64}$	00.000101
1	0	1	0	$\frac{17}{64}$	00.010001
1	0	1	1	$-\frac{23}{64}$	00.011011
1	1	0	0	$\frac{109}{64}$	01.101011
1	1	0	1	$\frac{69}{64}$	01.000101
1	1	1	0	$\frac{81}{64}$	01.010001
1	1	1	1	$\frac{41}{64}$	00.101001

- (e) Independent of the distributed arithmetic realization, realize the multiplication of coefficient  $a$  using additions, subtractions, and shifts with as few additions/subtractions as possible. (2 p)

**Solution:**

$$\frac{45}{64}X = \frac{5}{4} \times \frac{9}{8}X \gg 1$$

With

$$Y = X + X \gg 3$$

$$\frac{45}{64}X = (Y + Y \gg 2) \gg 1$$

where  $\gg$  denotes right-shift.