

# Written examination TSTE87

2014-06-03

1. (a) What is the difference between retiming and pipelining? (2 p)

**Solution:** Pipelining introduces delay elements between the input and the output, while retiming just moves the existing ones around.

- (b) What are the reasons for scaling signal levels? (Select exactly two reasons) (2 p)

**Solution:** Avoid overflow and reduce round-off noise.

- (c) What are the reasons for scheduling over more than one sample period? (Select exactly two reasons) (2 p)

**Solution:** Two out of:

- To obtain a more efficient resource allocation
- When the longest execution time is longer than the sample period
- If the sample period is not an integer multiple of the time unit used

- (d) How are the iteration period bound  $T_\infty$ , minimal sample period,  $T_{\min}$ , number of operations and number of delays respectively affected by unfolding an algorithm a factor  $N$ ? (2 p)

**Solution:**

- Iteration period bound: increases a factor  $N$
- Minimal sample period: unchanged
- Number of operations: increases a factor  $N$
- Number of delays: unchanged

- (e) Name two redundant number systems. (2 p)

**Solution:** For example, signed-digit representation and carry-save representation.

2. In a (simplified) interleaver for a communication system, the following sample re-ordering should happen. The input samples arrive in the order 1 to 9, while the output order is specified as: 3, 6, 2, 1, 5, 9, 8, 4, 7. This means that the first sample to exit the interleaver is input sample 3, then input sample 6, and so on, one per cycle. The resulting structure should use single port memories, i.e., in each cycle

a sample can either be read from or written to a memory. It is OK to bypass the memories, i.e., directly send a sample from the input to the output. Note that the interleaver shall be operated in a **cyclic/pipelined** manner, i.e., the next block of samples should be processed as soon as possible (it is possible to have continuous operation).

(a) Draw a memory life time graph for the interleaver.

(4 p)

**Solution:**

Input order/time	Output order	Output time
1	4	8
2	3	7
3	1	5
4	8	12
5	5	9
6	2	6
7	9	13
8	7	11
9	6	10

The offset between output order and output time is determined by the minimum required, for this case input 6. This also leads to that input 6 can be bypassed in the realization and does not have to be stored.

1

2

3

4

5

7

8

9

(b) Divide the variables between a minimal number of memories.

(4 p)

**Solution:**

(c) For each memory, assign the variables to a minimal number of memory cells.

(4 p)

**Solution:** Memory 1

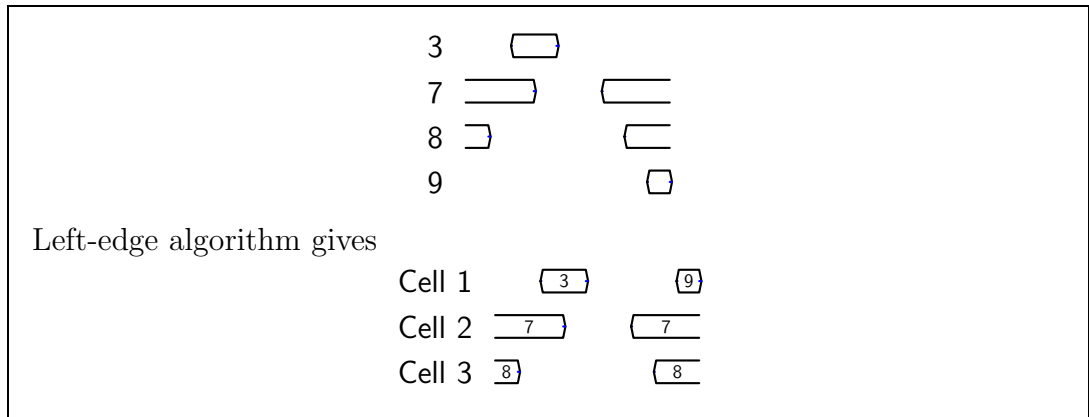
Cell 1

Cell 2

Cell 3

Cell 4

Memory 2



3. Some FPGAs have a DSP block consisting of an adder connected to a multiplier connected an adder, such that one can perform a computation of the type  $(a \pm b) \times c \pm d$ . It is also possible to obtain the result after the multiplication at the same time. This DSP block can be pipelined in four levels, one level after the first adder, one within the multiplier, one after the multiplier and one after the final adder. We will implement a symmetric two-port adaptor using the DSP block and a separate adder.

A symmetric two-port adaptor has the input/output relations

$$\begin{aligned}
 B_1 &= -\alpha A_1 + (1 + \alpha)A_2 \\
 B_2 &= (1 - \alpha)A_1 + \alpha A_2
 \end{aligned}$$

and can be realized using three adders/subtractors and one multiplier (coefficient  $\alpha$ ).

Draw the structure where shimming delays are correctly inserted assuming that all pipeline registers in the DSP block are used. (6 p)

4. The following expression shall be realized using distributed arithmetic. Determine the ROM contents in a suitable (stated) binary representation.

$$y = \frac{7}{16}x_1 - \frac{5}{8}x_2 + \frac{27}{32}x_3$$

(4 p)

<b>Solution:</b>					
$x_1$	$x_2$	$x_3$	Expression	Value	Two's complement
0	0	0	0	0	00.00000
0	0	1	$\frac{27}{32}$	$\frac{27}{32}$	00.11011
0	1	0	$-\frac{5}{8}$	$-\frac{5}{8}$	11.01100
0	1	1	$\frac{27}{32} - \frac{5}{8}$	$\frac{7}{32}$	00.00111
1	0	0	$\frac{7}{16}$	$\frac{7}{16}$	00.01110
1	0	1	$\frac{7}{16} + \frac{27}{32}$	$\frac{41}{32}$	01.01001
1	1	0	$\frac{7}{16} - \frac{5}{8}$	$-\frac{3}{16}$	11.11010
1	1	1	$\frac{7}{16} + \frac{27}{32} - \frac{5}{8}$	$\frac{21}{32}$	00.10101

5. The digital filter algorithm in Fig. 1 shall be implemented on a shared-memory architecture with homogeneous non-preemptive processing elements. The processing element has one input and contains a programmable shift, an accumulator. Hence, for an addition, the latency and execution time are both two clock cycles and the inputs should arrive at two different clock cycles (and, hence, the latency of one of the adder inputs is only one clock cycle). For a multiplication, the number of non-zero terms in the SD representation of the coefficient determines the latency and execution time. The multiplier coefficient values are

$$\begin{aligned} a &= \frac{3}{8} \\ b &= -\frac{11}{16} \\ c &= \frac{45}{64} \end{aligned}$$

- (a) Determine a minimal signed-digit representation for the coefficients and therefore the corresponding latencies of the multipliers. (2 p)

**Solution:**

$$\begin{aligned} \frac{3}{8} &= 0.10\bar{1} = 0.011 \Rightarrow 2 \text{ clock cycles} \\ -\frac{11}{16} &= \bar{1}.0101 = 0.\bar{1}\bar{1}01 = 0.\bar{1}0\bar{1}\bar{1} \Rightarrow 3 \text{ clock cycles} \\ \frac{45}{64} &= 1.0\bar{1}0\bar{1}01 = 0.110\bar{1}01 = 0.110011 = 1.0\bar{1}0011 \Rightarrow 4 \text{ clock cycles} \end{aligned}$$

- (b) Determine the minimal sample period,  $T_{\min}$ . Note that the latency of the two inputs to the addition are different. (2 p)

**Solution:** The result will depend on which order the inputs of the adders are assigned. The following assignment gives the best result.

Now,

$$T_{\min} = \max \left\{ \frac{2+1}{1}, \frac{3+2+1+1}{2} \right\} = 3.5 \text{ clock cycles.}$$

Note that changing the order of the middle adder gives

$$T_{\min} = \max \left\{ \frac{2+2}{1}, \frac{3+1+1+1}{2} \right\} = 4 \text{ clock cycles.}$$

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

**Solution:** With the adder order above, the critical path is

$$T_{cp} = 4 + 2 + 3 = 9 \text{ clock cycles.}$$

- (d) Determine the precedence graph. (6 p)

- (e) Schedule the algorithm for  $T_s = 5$  clock cycles and a reasonably efficient resource utilization. Make sure that you indicate all the connections in the schedule. (8 p)
- (f) How many processing elements are required? What is the theoretical minimum? Comment on any difference. (2 p)

**Solution:** The theoretical bound is

$$N_{PE} = \left\lceil \frac{\sum T_{exe}}{T_{schedule}} \right\rceil = \left\lceil \frac{3 \times 2 + 2 + 3 + 4}{5} \right\rceil = 3.$$

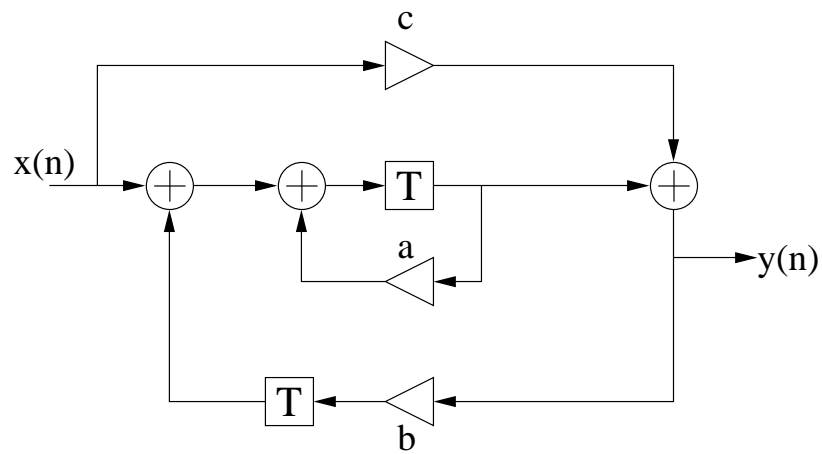


Figure 1: Digital filter for use in Problem 5.

6. Consider the implementation of a 1024-point FFT in an FPGA. The implementation should perform an FFT in  $12 \mu\text{s}$  and will be implemented in one of two different ways. In the FPGA we can use memories, multipliers and adders operating at 300 MHz. The memories can read two samples and write one in a cycle. All samples are complex values.

Recall that the number of butterfly operations (with the definition below) is  $\frac{N}{2} \log_2(N)$  for an  $N$ -point FFT.

- (a) Consider an implementation where the PE is a butterfly consisting of a radix-2 DFT stage (one adder and one subtracter) followed by a multiplier. How many processing elements and memories are required for the implementation? (4 p)

**Solution:** The number of butterfly operations per second is

$$\frac{\frac{1024}{2} \log_2(1024)}{12 \times 10^{-6}} = \frac{1280}{3} \times 10^6$$

With the butterfly PE operating at 300 MHz this gives that

$$\frac{\frac{1280}{3} \times 10^6}{300 \times 10^6} = \frac{64}{45} \approx 1.42 \Rightarrow 2$$

adaptor PEs are needed.

The number of memory accesses are for both reading and writing two times the number of adaptor operations. Since writing is more limited this gives that

$$2 \frac{64}{45} \approx 2.84 \Rightarrow 3$$

memories are required to cope with the data flow. It is also OK to use the fact that there are two adaptors leading to four memories. However, this will affect the answer to sub problems (c) and (d).

- (b) Instead consider implementing the FFT using real-valued operations, i.e., a complex valued adder is two real-valued adders and a complex valued multiplier is four real-valued multipliers and two real-valued adders. How many adders, multipliers and memories are required for the implementation? (4 p)

**Solution:** There are 6 real-valued additions and 4 real-valued multiplications performed in each adaptor operation. Hence, the numbers of additions and multiplications per second are

$$6 \times \frac{1280}{3} \times 10^6 = 2560 \times 10^6$$

and

$$4 \times \frac{1280}{3} \times 10^6 = \frac{5120}{3} \times 10^6$$

respectively. This leads to that the numbers of adders and multipliers required are

$$\frac{2560 \times 10^6}{300 \times 10^6} = \frac{128}{15} \approx 8.53 \Rightarrow 9$$

and

$$\frac{\frac{5120}{3} \times 10^6}{300 \times 10^6} = \frac{256}{45} \approx 5.69 \Rightarrow 6$$

respectively.

The amount of data to be read per second is

$$2 \times 2560 \times 10^6 + \frac{5120}{3} \times 10^6 = \frac{20480}{3} \times 10^6$$

and to be written

$$2560 \times 10^6 + \frac{5120}{3} \times 10^6 = \frac{12800}{3} \times 10^6$$

This leads to that the writing is more limiting, so the number of memories required is

$$\frac{\frac{12800}{3} \times 10^6}{300 \times 10^6} = \frac{128}{9} \approx 14.22 \Rightarrow 15$$

The same number can be obtained by realizing that the PEs should access  $2 \times 9 + 6$  inputs and write  $9 + 6 = 15$  outputs.

- (c) Translate the number of complex operations and memories of the first approach to the corresponding real-valued numbers and compare the two architectures in terms of implementation and control complexity. (2 p)

**Solution:**

Approach	Multipliers	Adders	Memories
Butterfly PE	$2 \times 4 = 8$	$2 \times 6 = 12$	$3(4) \times 2 = 6(8)$
Separate operations	6	9	15

It can be seen that more operators are needed for the butterfly approach, but that the number of memories is decreased. One would expect simpler control of the butterfly approach since there are fewer read and write addresses to be generated in each cycle. In addition, as that approach is based on complex numbers it will be even simpler as only 6(8) (read and write for 3(4) memories) addresses are required.

- (d) Are the numbers you have come up with realistic/practical in the sense that it should be rather straightforward to implement an architecture using this number of PEs and memories? Comment and if needed suggest improvement for a more practical solution. (2 p)

**Solution:** For the first approach it will make more sense to use four memories to obtain an easier implementation. Although it is enough to use three, the access of these would have to be scheduled in time in a non-trivial way using intermediate cache memories and would, hence, most likely result in a more complicated overall solution.

For the second case, the main challenge will be to handle the memory and avoid memory conflicts.

7. Unfold the digital filter SFG in Fig. 2 a factor of 2.

(4 p)

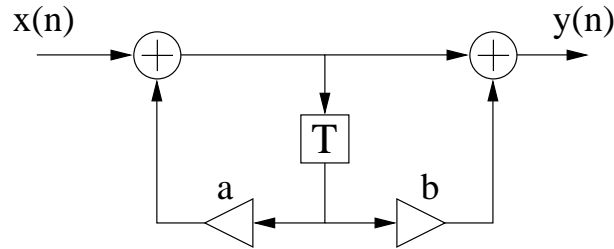
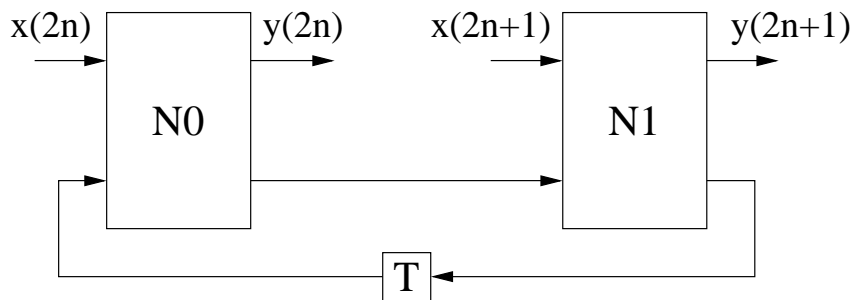


Figure 2: Digital filter for use in Problem 7.

**Solution:**

General unfolding



Specific problem

