

Written Examination TSTE87

2017-05-29

1. (a) Give two examples of use cases where a dedicated implementation may be advantageous compared to using general purpose DSP processors. (2 p)

Solution:

- (b) Define latency and execution time. (2 p)

Solution: Latency: time from an input to the corresponding output.
Execution time: time from an input to the next input.

- (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)

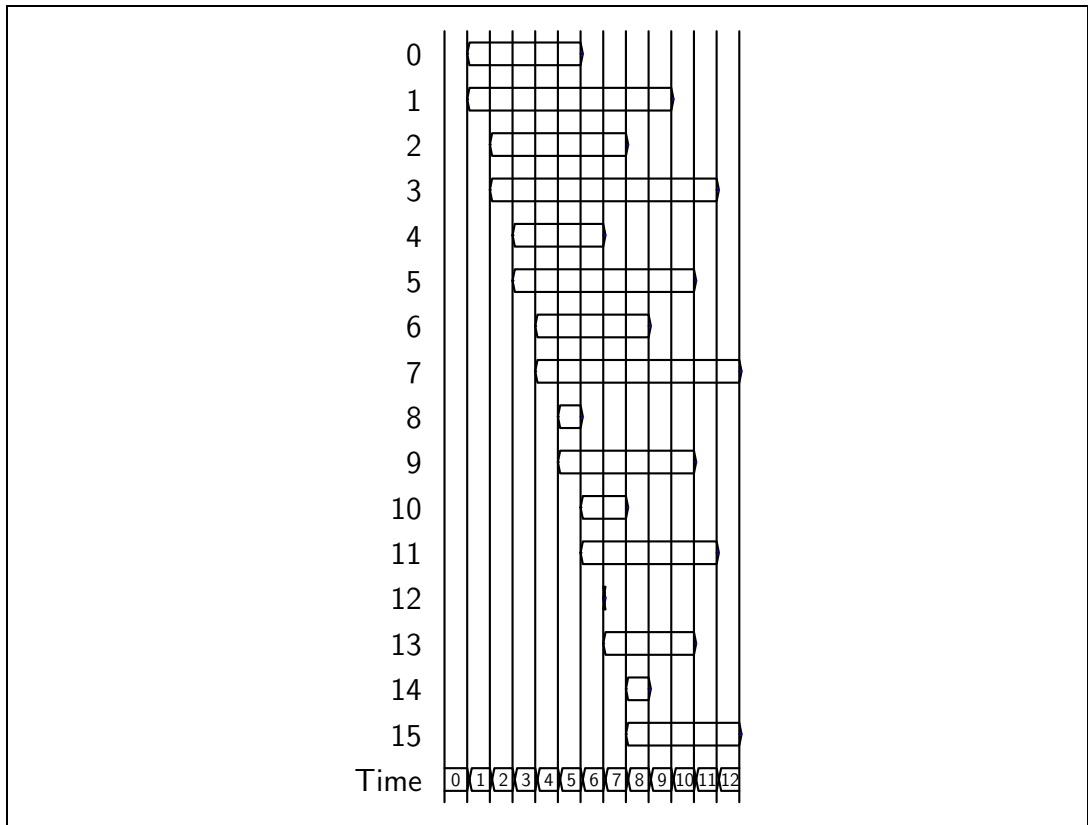
Solution:

- (e) Name and describe one way to accelerate a carry-propagation adder. (2 p)

2. In an FFT, there is a need for bit-reversal of a sequence consisting of 16 samples, arriving two at a time. Hence, in time slot 1, samples 0 and 1 arrives, in time slot 2, samples 2 and 3 and so on. The outputs should be in bit-reversed order, so the first samples to leave the bit-reversal circuits is sample 0 (=0000 bit-reversed) and sample 8 (=0001 bit-reversed). After that, sample 4 (=0010 bit-reversed) and sample 12 (=0011 bit-reversed) should leave the circuit, and so on.

- (a) Draw a memory life time graph for the bit-reversal circuit. (6 p)

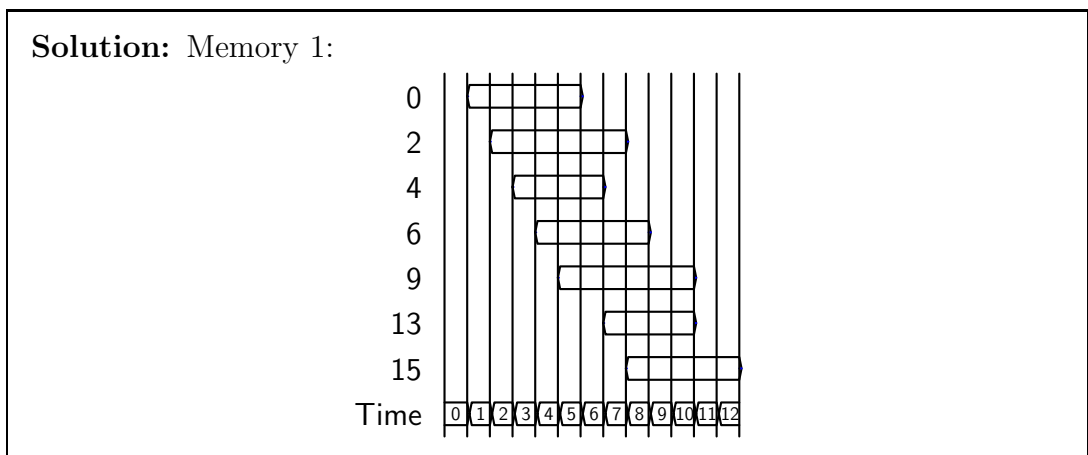
	Sample	Input order/time	Output order	Output time
	0	1	1	6
	1	1	5	10
	2	2	3	8
	3	2	7	12
	4	3	2	7
	5	3	6	11
	6	4	4	9
Solution:	7	4	8	13
	8	5	1	6
	9	5	5	10
	10	6	3	8
	11	6	7	12
	12	7	2	7
	13	7	6	11
	14	8	4	9
	15	8	8	13

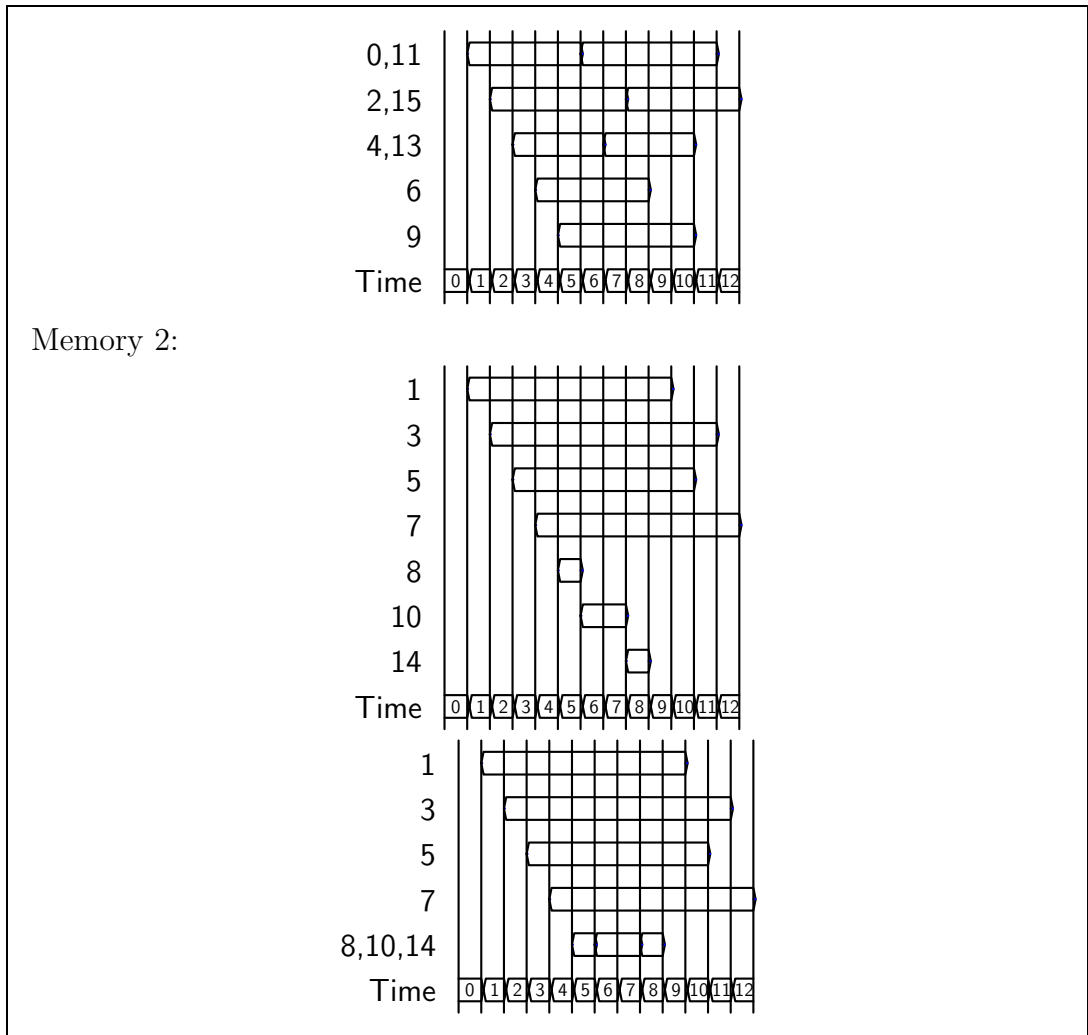


- (b) Divide the variables between a minimal number of memories. For each memory it is possible to read one variable and write one variable in each cycle. (4 p)

Solution: Skip 12 that can go directly from input to output. Variables $\{0, 9\}$ can be in the same memory, different from $\{1, 8\}$. Same for $\{2, 11\}$, do not share with $\{3, 10\}$. $\{4, 13\}$, $\{5\}$. $\{6, 15\}$, $\{7, 14\}$.
 Select e.g. memory 1: $\{0, 2, 4, 6, 9, 11, 13, 15\}$, memory 2: $\{1, 3, 5, 7, 8, 10, 14\}$.

- (c) For each memory, assign the variables to a minimal number of memory cells. It is possible to read from and write to the same memory position in a cycle. (4 p)





3. Consider the digital filter algorithm in Fig. 1.

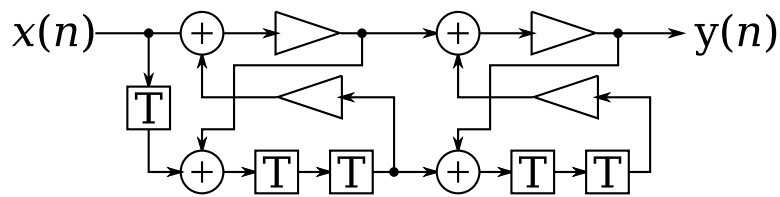


Figure 1: Digital filter for use in Problems 3.

(a) Determine the minimal sample period, T_{\min} .

(2 p)

Solution:

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

(b) Determine the critical path, T_{cp} .

(2 p)

Solution:

$$T_{cp} = 3T_{add} + 3T_{mult}$$

- (c) Apply pipelining and retiming to obtain an algorithm with $T_{cp} = T_{min}$. (4 p)

4. The digital filter algorithm in Fig. 2 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 two clock cycles and the latency of an adder is one clock cycle. Both types of processing elements have an execution time of two clock cycles.

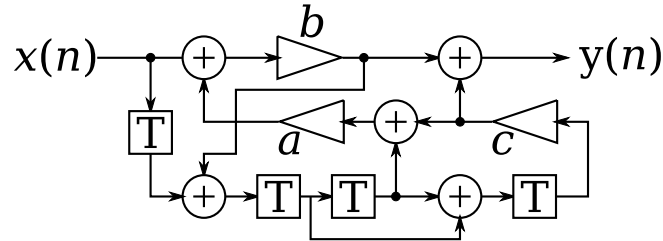


Figure 2: Digital filter for use in Problems 4 and 5.

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

Solution:

$$T_{\min} = \max \left\{ \frac{3T_{\text{add}} + 2T_{\text{mult}}}{2}, \frac{4T_{\text{add}} + 3T_{\text{mult}}}{2}, \frac{4T_{\text{add}} + 3T_{\text{mult}}}{3} \right\}$$

$$= 2T_{\text{add}} + \frac{3}{2}T_{\text{mult}} = 5 \text{ clock cycles}$$

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

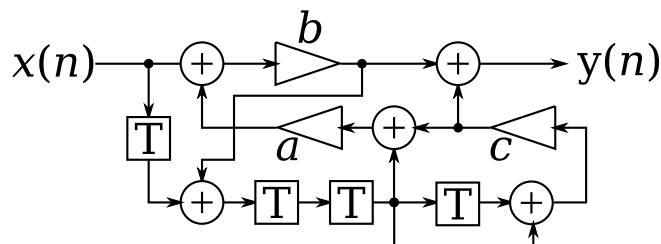
Solution:

$$T_{cp} = 3T_{\text{add}} + 3T_{\text{mult}} = 9 \text{ clock cycles}$$

- (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)
5. The filter in Fig. 2 is now to be implemented using distributed arithmetic. The filter coefficients are, $a = -\frac{5}{8}$, $b = -\frac{5}{4}$, and $c = \frac{3}{4}$.

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

Solution:



- (b) Determine the state-space representation of the retimed algorithm. (4 p)

Solution:

$$\begin{bmatrix} v_1(n+1) \\ v_2(n+1) \\ v_3(n+1) \\ v_4(n+1) \\ y(n) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & ab(1+c) & abc & b \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & ab+abc+c & abc+c & b \end{bmatrix} \begin{bmatrix} v_1(n) \\ v_2(n) \\ v_3(n) \\ v_4(n) \\ x(n) \end{bmatrix}$$

- (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)
6. Consider the implementation of a spectrometer for radio astronomy. Such a system consists of a long length FFT and for each FFT computation, the each individual FFT output (a bin/tap) is squared and accumulated to the sum of squared taps from the previous FFT. This means that for an N -point FFT, there will be N accumulated sums of squares that must be stored. The spectrometer should be implemented in an FPGA with 2016 multipliers and 38304 kb of memory. The amount of memory required to compute an N -point FFT is $\approx N$ complex samples. Assume a data word length of 18 bits (18 bits real and 18 bits imaginary), both for the FFT computation and the sum of squares, that a complex multiplication requires four multipliers, a complex square two multipliers, and a clock frequency of 500 MHz for the FPGA.
- (a) Determine the maximum FFT length (power of two) assuming a data rate of 4 GSa/s. (4 p)

Solution:

$$\text{Multiplications per sample} \approx \left(\underbrace{\frac{4\frac{N}{2} \log_2(N)}{N}}_{\text{FFT}} + \underbrace{\frac{2N}{N}}_{\text{Sum of squares}} \right) \frac{f_{\text{sample}}}{f_{\text{clk}}} \leq 2016$$

$$(2 \log_2(N) + 2) \frac{f_{\text{sample}}}{f_{\text{clk}}} \leq 2016$$

$$16 \log_2(N) + 16 \leq 2016 \Rightarrow N \leq 2^{125}$$

$$\text{Memory} \approx \left(\underbrace{2N}_{\text{FFT}} + \underbrace{N}_{\text{Sum of squares}} \right) 18 \leq 38304 \times 10^3$$

$$N \leq \frac{38304 \times 10^3}{3 \times 18} \Rightarrow N \leq 2^{19}$$

- (b) Depending on if the number of multipliers or amount of memory was limiting in the previous question: where is the break-even point in terms of data rate for the same FFT length? That is, the data rate where both the memory and the multipliers are fully used (within reasonable limits). (2 p)

$$\text{Solution: } \frac{f_{\text{sample}}}{f_{\text{clk}}} \leq \frac{2016}{2 \times 19 + 2} \Rightarrow f_{\text{sample}} = 25.2 \text{ GSa/s}$$