Written examination TSTE87

2015-06-03

1. (a) Why is the latency of a bit-serial (LSB first) multiplier at least equal to the number of fractional bits? (2 p)

   Solution:

   (b) Why are only the loops included when determining the minimal sample period? (And, e.g., not the critical path?) (2 p)

   Solution: Non-recursive parts can be pipelined/retimed.

   (c) How is the throughput (number of operations per time unit) related to the execution time? (2 p)

   Solution:

   $$\text{Throughput} = \frac{1}{T_{exe}}$$

   (d) What is a redundant number representation? Give one example. (2 p)

   Solution: In a redundant number representation there is more than one representation of each number.

   (e) How are the iteration period bound, T_∞, minimal sample period, $T_{\min}$, number of operations and number of delay elements 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 elements: unchanged

2. Consider the memory variable life time diagram in Fig. 1, consisting of variables a to h. This should be realized using single port memories, i.e., memories where a single variable can be either read or written in each time slot.

   (a) Divide the variables between a minimal number of memories. (4 p)

   Solution: Exclusion graph based on memory read/writes

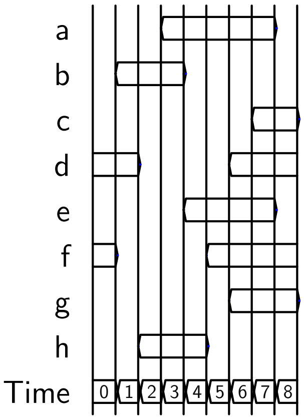
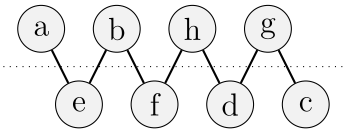


Figure 1: Life time diagram for Problem 2.

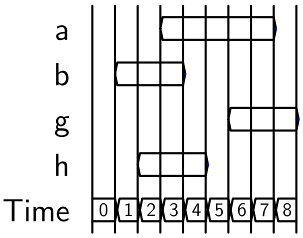


so {a, b, g, h} in one memory and {c, d, e, f} in the other memory.

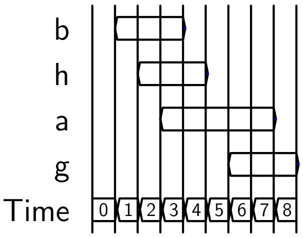
(b) For each memory, assign the variables to a minimal number of memory cells. (4 p)

Solution: Use the left edge algorithm for each memory.

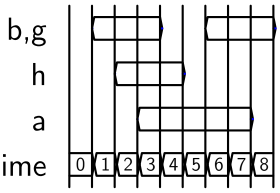
Memory 1:



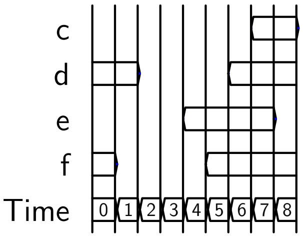
Sorted:



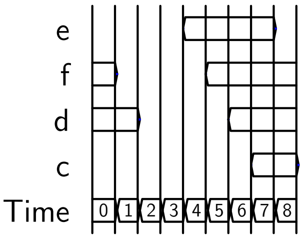
Assigned:



Memory 2:



Sorted:



No sharing possible.

3. The following expression shall be realized using shifts, additions and subtractions. Draw a realization using as few additions and subtractions as possible.

$$y = 13x_1 - 5x_2 + 25x_3$$

Note that $25 = 5 \times 5 = 13 \times 2 - 1$. (4 p)

Solution:

$y = 13x_1 - 5x_2 + 25x_3 = 8x_1 + 5(x_1 - x_2 + 5x_3) = 8(x_1 + 2x_3) + 5((x_1 + 2x_3) - x_2) - x_3$

so five additions/subtractions are required.

4. The expression in Problem 3 is now to be implemented using distributed arithmetic. Determine the ROM contents in a named and stated binary representation. (4 p)

Solution: Two's complement representation

x_1	x_2	x_3	Value	ROM content
0	0	0	0	0000000
0	0	1	25	0011001
0	1	0	-5	1111011
0	1	1	20	0010100
1	0	0	13	0001101
1	0	1	38	0100110
1	1	0	8	0001000
1	1	1	33	0100001

5. Consider the realization of a complex-valued 40-tap (39:th-order) FIR filter in an FPGA. The FPGA can be clocked at a maximum of 500 MHz and the input sample rate is 2.5 GSa/s. For simplicity, assume that each complex multiplier requires four multipliers and that no coefficient symmetry can be used.

   (a) From the following list of FPGAs, which one (a, b, c, or d) would be the most suitable for a direct implementation of the FIR filter, i.e., the design fits but the FPGA is not overly large? Each DSP block contains one multiplier. (4 p)

	FPGA	DSP blocks
a	XC6VLX75T	288
b	XC6VLX130T	480
c	XC6VLX195T	640
d	XC6VSX315T	1344

Solution:

$$\frac{40 \times 4 \times 2.5 \times 10^9}{500 \times 10^6} = 800 \text{ multipliers} \tag{1}$$

so FPGA d must be selected.

   (b) Now, consider the implementation of the FIR filtering in the frequency domain, i.e., using a DFT/FFT. For an N-point DFT, in this case, $N - 40 + 1$ samples can be processed per iteration (samples + impulse response length - 1 = DFT length). Each iteration require the computation of an N-point DFT, N complex multiplications and an N-point IDFT (identical complexity to DFT). The computation of an N-point FFT (with $N = 2^M$) require $\frac{N}{2}(\log_2 N - 1) = \frac{N}{2}(M-1)$ complex multiplications. Assuming a "perfect" resource utilization, i.e., no worrying about scheduling the algorithms etc, which is the most suitable FPGA? Hint: the optimal complexity (in multiplications per sample processed) is found for about four times more samples than impulse response length for each iteration. (6 p)

Solution: Each DFT-mult-IDFT iteration require

$$\left(\frac{N}{2}(\log_2 N - 1) + N + \frac{N}{2}(\log_2 N - 1)\right) = N \log_2 N \text{ complex multiplications}$$

and will process $N - 39$ samples. Hence, the number of complex multiplications per sample is

$$\frac{N \log_2 N}{N - 39}$$

Complex multiplications per sample, $N = 64$

$$\frac{64 \times 6}{25} = 15.36 \tag{2}$$

Complex multiplications per sample, $N = 128$

$$\frac{128 \times 7}{89} \approx 10.07 \tag{3}$$

Complex multiplications per sample, $N = 256$

$$\frac{256 \times 8}{217} \approx 9.44 \tag{4}$$

Complex multiplications per sample, $N = 512$

$$\frac{512 \times 9)}{473} \approx 9.74 \tag{5}$$

Best result for $N = 256$. Number of multipliers needed:

$$\frac{9.44 \times 4 \times 2.5 \times 10^9}{500 \times 10^6} \approx 189 \tag{6}$$

so FPGA a can be used instead.

(c) Assuming that you utilize all multipliers in the selected FPGAs for (a) and (b), what are the respective resulting clock frequencies required? (2 p)

Solution: For (a) FPGA d is selected which have 1344 DSP blocks compared to the 800 needed. Hence, the new clock frequency is $500 \times \frac{800}{1344} \approx 298$ MHz. For (b) FPGA a is selected which have 288 DSP blocks compared to the 189 needed. Hence, the new clock frequency is $500 \times \frac{189}{288} \approx 328$ MHz.

6. Consider the unfolded first-order filter in Fig. 2.

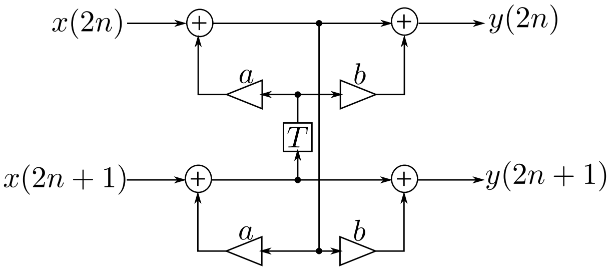


Figure 2: Unfolded first-order filter.

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

Solution:

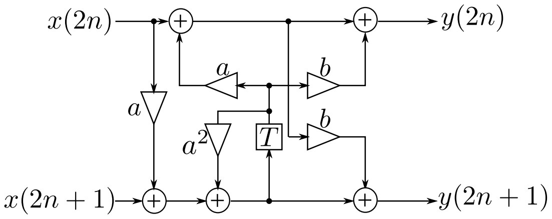
$$T_{\min} = \frac{T_\infty}{2} = \frac{\frac{2T_{L,mult}+2T_{L,add}}{1}}{2} = \frac{2T_{L,mult} + 2T_{L,add}}{2} = T_{L,mult} + T_{L,add}$$

(b) Apply algorithm transformations and/or pipelining/retiming to obtain $T_{cp} = T_\infty = T_{L,mult} + T_{L,add}$, i.e., there should be at most one multiplier and one adder in the critical path and critical loop, respectively. Ignore finite word length effects. (6 p)

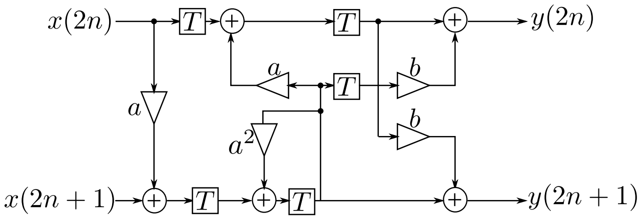
Solution:

$$\begin{cases} y(2n) &= x(2n) + (a+b)v(n) \\ y(2n+1) &= (a+b)x(2n) + x(2n+1) + a(a+b)v(n) \\ v(n+1) &= ax(2n) + x(2+1) + aav(n) \end{cases}$$

Focus on the loop, merge the two multiplications by a and reorder the additions:



Pipeline and retime:



7. In Fig. 3, the block diagram of a 5-point DFT based on Winograd short-DFT algorithm is shown. This should be implemented using a time-multiplexed shared-memory architecture with homogeneous non-preemptive processing elements. There are two types of processing elements: multipliers and adders (which can also perform subtractions). All signals are complex-valued, however, the processing elements are also complex-valued. 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 one clock cycle. The implementation should be operated in a cyclic manner, i.e., you can pipeline the block diagram arbitrarily. In one scheduling period, the inputs should arrive one per time units in order $x(0),\ x(1),\ldots$ and outputs exit one per time unit in order $X(0), X(1), \ldots$, note that this can be readily obtained using shimming delays.

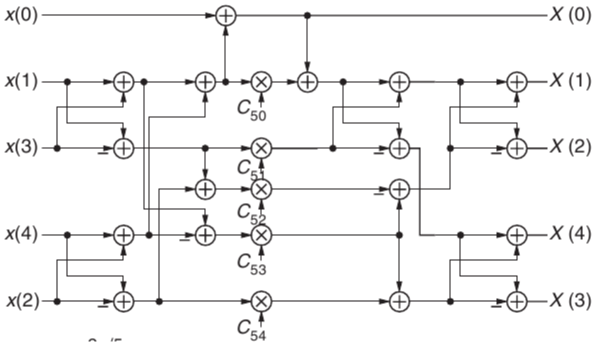


Figure 3: 5-point Winograd DFT.

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

Solution:

$$T_{cp} = T_{L,mult} + 5T_{L,add} = 7 \text{ clock cycles}$$

(b) Determine the precedence graph. (6 p)

(c) Determine the minimum scheduling period assuming that one multiplier PE is used. (Ignore the additions for now.) (2 p)

Solution:

$$N_{\text{PE}} = \frac{\sum T_{\text{exe}}}{T_{\text{schedule}}} \Rightarrow T_{\text{schedule}} = \frac{\sum T_{\text{exe}}}{N_{\text{PE}}} = \frac{5 \times 1}{1} = 5 \text{ time units} \tag{7}$$

(d) Based on the scheduling period from (a) determine the minimum amount of adder PEs. (2 p)

Solution:

$$N_{\text{PE}} = \frac{\sum T_{\text{exe}}}{T_{\text{schedule}}} = \frac{17 \times 1}{5} = 3.4 \text{ adder PEs} \Rightarrow 4 \text{ adder PEs} \tag{8}$$

(e) Schedule the algorithm using one multiplier, a minimum amount of adder PEs and the minimum scheduling period. It is OK to drop the signs from the subtractions. (8 p)

(f) What is the latency of the resulting schedule (from input $x(i)$ to output $X(i)$)? Compare to the critical path, T_{cp}. (2 p)

(g) A direct computation of the DFT requires multiplying the input vector by a 5×5 complex-valued DFT matrix. What are the savings in additions/subtractions and multiplications, respectively, using the Winograd 5-point DFT compared to a direct computation? Note that for the first row and the first column of the DFT matrix, the coefficients are 1, so those multiplications should be ignored. (2 p)

Solution:

	Additions/subtractions	Multiplications
Direct computation	$5 \times (5-1) = 20$	$(5-1) \times (5-1) = 16$
Winograd	17	5
Savings	3	11
Relative savings	15%	68.75%