Homework #3

This homework consists of questions taken from the notes and open-ended questions. You are assigned as groups to do the homework. All problems are graded on CC scale (Completion only).

1. (10 points, CC) Consider the block diagram DFG below:

Assume that the addition and multiplication require 1 and 2 units of time respectively.

(a) (3 points, CC) What is the iteration bound $T_\infty$?
(b) (7 points, CC) Find the minimum unfolding factor $J$ such that the $J$-unfolded DFG can be retimed so that the critical path of this unfolded and retimed DFT is $JT_\infty$.

2. (15 points, CC) Consider the DFG shown below where the computation time of each node is labeled in parentheses:

(a) (3 points, CC) What is the iteration bound $T_\infty$?
(b) (5 points, CC) Retime this DFG to minimize iteration period.
(c) (7 points, CC) Determine the minimum unfolding factor $J$ such that the $J$-unfolded DFG (unfold from the original DFG) can be retimed so that the critical path of this unfolded and retimed DFG is $JT_\infty$ where $T_\infty$ is determined in part (a). Also plot the unfolded and retimed DFG and identify its critical path whose delay is $JT_\infty$.

3. (10 points, CC) Consider the iteration DG given below:

(a) (3 points, CC) What is the iteration bound $T_\infty$?
(b) (5 points, CC) Retime this DFG to minimize iteration period.
(c) (7 points, CC) Determine the minimum unfolding factor $J$ such that the $J$-unfolded DFG (unfold from the original DFG) can be retimed so that the critical path of this unfolded and retimed DFG is $JT_\infty$ where $T_\infty$ is determined in part (a). Also plot the unfolded and retimed DFG and identify its critical path whose delay is $JT_\infty$. 


For each pair of scheduling vector \( s \) and projection (assignment) vector \( d \), (i) determine if they are permissible, (ii) if they are permissible, compute the processor space \( P \), and perform the node mapping, arc mapping, and I/O mapping.

(a) (6 points, CC) \[
\begin{bmatrix}
1 \\
0
\end{bmatrix}; \quad \begin{bmatrix}
1 \\
0
\end{bmatrix}
\]

(b) (4 points, CC) \[
\begin{bmatrix}
1 \\
-1
\end{bmatrix}; \quad \begin{bmatrix}
1 \\
0
\end{bmatrix}
\]

4. (20 points, CC) Consider the correlation of two 1-dimensional sequences \( \{x(n)\} \) and \( \{y(n)\} \) with \( 0 \leq n \leq N-1 \).

\[
r(n) = \sum_{k=0}^{N-1} x(k)y(k+n)
\]

(a) (5 points, CC) Write a C program (or use any high level language) source code that compute \( \{r(n)\} \) for \( N=5 \), and \( n = 0, 1, 2 \).

(b) (5 points) Convert this program into single assignment format.

(c) (5 points) Convert this program so that it has local data communications.

(d) (5 points, CC) Plot the corresponding iterative DG.

5. (20 points, CC) (Binary number multiplication) Consider two unsigned, \( n \)-bit binary numbers \( A = A_{n-1}A_{n-2} \ldots A_1A_0 \), and \( B = B_{n-1}B_{n-2} \ldots B_1B_0 \), where \( A_i, B_i \in \{0, 1\}, 0 \leq i \leq n-1 \). Denote \( C = A \times B = C_{2n-2}C_{2n-3} \ldots C_1C_0 \) to be an \( 2n-1 \) bit binary number representing the product of \( A \) and \( B \).

(a) (4 points, CC) Derive an mathematical expression of \( C_k \) in terms of \( A_i \)s and \( B_i \)s.

(b) (3 points) Write a C or Matlab® program as a nested loop construct to compute \( C_k \) for \( 0 \leq k \leq 2n-2 \). Make sure this nested loop has single-assignment format. Use variable indices as \( A(I), B(I), C(K) \), etc.

(c) (3 points) Localize the program in part (b) by converting all variables that need to be broadcast (used by more than one iterations) during the execution into transmittal variables that will be passed along from an iteration to the next.

(d) (3 points, CC) Plot the logic schematic diagram of the architecture to implement the operations of the loop body as specified in part (c). Note that \( A(I), B(I) \) are both binary variables.
(e) (3 points) For \( n = 4 \), plot the DG of the localized program in part (c), and identify the critical path of this DG. A hardware implementation of this DG is called an array multiplier.

(f) (4 points) Use cut-set retiming to convert the DG directly into a 2D systolic array that implements a fully pipelined version of the array multiplier.

6. (10 points, CC) Saturation arithmetic

Saturation arithmetic is a convenient, nonlinear method of handling overflow of fixed-point computation in DSP applications. Based on the types of two operands and one result, there are 8 possible combinations. In practice, the three commonly used options are sss, uuu, and uus where \( u \) stands for unsigned, and \( s \) stands for signed. In Intel MMX and SSE-2 instruction set, only uuu, and sss are implemented. The former is called unsigned saturation arithmetic, and the latter is called signed saturation arithmetic. Suppose now there are two packed operands each consists of four 16-bit words as shown below

\[
R_a: \begin{bmatrix} 58 & 14 & 12 & 77 \end{bmatrix}
\]

\[
R_b: \begin{bmatrix} 22 & 192 & 118 & 36 \end{bmatrix}
\]

The content is given in decimal format for convenience. Refer to MMX instruction set,

(a) (5 points, CC) Write a MMX assembly code segment to compute the maximum of each pair of integers at the corresponding position. Assuming the result is stored in register \( R_c \). The final content of \( R_c \) should be

\[
R_c: \begin{bmatrix} 58 & 192 & 118 & 77 \end{bmatrix}
\]

You should prove that your program is correct. (Hint: Use unsigned saturation arithmetic)

(b) (5 points, CC) Still given \( R_a \) and \( R_b \) with their original content shown in part (a) of this problem. Now, our goal is to perform absolute difference between each pair of the words in \( R_a \) and \( R_b \). Using registers \( R_e, R_f, \ldots \) to store intermediate results. The final result should be stored in \( R_d \). The absolute difference between \( a \) and \( b \) is \(|a - b|\).

7. (15 points, CC)

Consider the RIA implementation of the FIR filter

\[
y_1(n,-1)=0, \quad n = 0,1,2, \ldots
\]

\[
h_1(0,k)=h(k), \quad k=0,\ldots,N-1
\]

\[
n=0,1,2,\ldots, \quad \text{and} \quad k=0,\ldots,N-1
\]

\[
y_1(n,k)=y_1(n,k-1)+h_1(n,k)\times x_1(n,k)
\]

\[
h_1(n,k)=h_1(n-1,k)
\]

\[
x_1(n,k)=x_1(n-1,k-1)
\]

\[
y(n)=y_1(n,N-1), \quad n=0,1,2,\ldots
\]

And the corresponding DG:
(a) (5 points, CC) Consider a uni-modular transformation matrix
\[
\begin{bmatrix}
1 & -1 \\
0 & 1
\end{bmatrix}
\]
Find the corresponding dependence matrix after this transformation,

(b) (5 points, CC) Plot the corresponding DG, and

(c) (5 points, CC) Write down the corresponding RIA formulation