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. (5 points)
   (a) (3 points) Consider the following Fortran-like program
   
   ```fortran
   DO I=1,N
       DO J=2,N
           A(I,J)=B(I,J)+C(I,J)
           C(I,J)=D(I,J)/2
           E(I,J)=A(I,J-1)**2+E(I,J-1)
       END
   END
   ```

   Assume that each array has already been initialized before executing this program. Rewrite the program into a vectorized format.

   (b) (2 points) Consider the following Fortran-like program
   
   ```fortran
   DO I=1,5
       A(I)=B(I)+I
       D(I)=A(I)+A(I+1)
   END
   ```

   Assume that arrays \{A(I)\} and \{B(I)\} have already been initialized before executing this program. Rewrite the program into a vectorized format.

2. (10 points) Consider the following Fortran-like program
   ```fortran
   C listing 1
   DO 10 I=1,3
       DO 10 J=a*I+b,c*I+d
       X(I,J)=X(I+2,J-1)+X(I-1,J)
   10 CONTINUE
   ```

   (a) (4 points) Let \(a\), \(b\), \(c\), and \(d\) be four constant integers. Find the condition(s) such that this nested loop is a regular nested loop.

   (b) (1 points) Derive the dependence matrix \(D\) of this algorithm.

   (c) (2 points) If \(a = c = 0\), \(b = 1\), and \(d = 2\). In order to execute this algorithm, which elements of the \(X\) array must be given as initial conditions?

   (d) (3 points) This algorithm as specified in part (c) may have a problem executing properly. Discuss what may be the problem and provide a reformulated program that can be executed correctly. Assuming all the initial conditions of the array \(X\) are available in the memory.

3. (12 points) Iteration bound problems
   (a) (3 points) Text book [Parhi], Chapter 2, problem 1.
   (b) (3 points) Text book [Parhi], Chapter 2, problem 2.
   (c) (3 points) Text book [Parhi], Chapter 2, problem 3.
   (d) (3 points) Text book [Parhi], Chapter 2, problem 4.
4. (12 points) Chapter 3 problems.
   (a) (3 points) Text book [Parhi], Chapter 3, problem 1.
   (b) (3 points) Text book [Parhi], Chapter 3, problem 2.
   (c) (3 points) Text book [Parhi], Chapter 3, problem 3. Hint: y(n) is the sum of two FIR filters.
   (d) (3 points) Text book [Parhi], Chapter 3, problem 5.

5. (31 points) Chapter 4 problems
   (a) (3 points) Text book [Parhi], Chapter 4, problem 1 (a), (b)
   (b) (4 points) Text book [Parhi], Chapter 4, problem 2.
   (c) (3 points) Text book [Parhi], Chapter 4, problem 3.
   (d) (4 points) Text book [Parhi], Chapter 4, problem 5.
   (e) (3 points) Text book [Parhi], Chapter 4, problem 7.
   (f) (4 points) Text book [Parhi], Chapter 4, problem 8.
   (g) (3 points) Text book [Parhi], Chapter 4, problem 10.
   (h) (7 points) Text book [Parhi], Chapter 4, problem 11.

6. (5 points) Refer to figure 4.20, page 115 of the text book [Parhi]. Assume the addition takes 2 t.u. and multiplication takes 5 t.u.
   (a) (1 points) Find the iteration bound \( T_\infty \).
   (b) (2 points) Identify the critical path \( P_{cr} \). Give the answer in the form of \( n_1 \rightarrow n_2 \rightarrow \ldots \) where \( n_1, n_2 \) are node numbers.
   (c) (2 points) In order to reduce the clock cycle time to the iteration bound, retiming are performed. Among many possible solutions, some of them will NOT add delays to the computation of the output (i.e. \( y(n) \) will be computed during the same clock cycle \( x(n) \) is sampled), and will NOT increase (decrease is OK) the total number of registers. Find two such solutions by giving their corresponding retimed DFGs.

7. (10 points) Down load the H.264 reference software jm10.2.zip from the web. Compile the program on a platform of your choice (Window or Unix). Run the test sequence at the location: \JM/bin/foreman_part_qcif.yuv\ on encoder first to produce encoded bit stream. Then run the decoder to decode the bit stream. Post the encoded bit stream and decoded video sequence on your class website.

8. (15 points) Motion estimation is the most computational intensive task in video coding. In this problem, we consider motion estimation over fixed 16 \times 16\ macro-block only. Assume that the video frame has a size of 640 (H) \times 480 (V)\ pixels (VGA), the frame rate is 15 frames per second. Also assume the video sequence is subsampled at 4:1:1 ratio among the YCbCr components.
   (a) (2 points) How many macro-blocks will need to be processed per second for the given frame size and frame rate?
   (b) (2 points) Assume that there is only a single reference frame used in motion estimation and the search range is \pm 8\ pixels. For each pixel in the reference frame, how many times it will be used for the motion estimation computation? Assume the pixel is at the interior of the reference frame, and hence the boundary situation can be ignored. Also assume that full-search algorithm is used.
   (c) (2 points) Repeat this problem for the case when the search range is increased to 64.
   (d) (2 points) Now, let us consider implementing the motion estimation on an embedded platform. Assume fixed 16 \times 16\ MB, \pm 8\ pixels search range, VGA frame size and 15 fps frame rate. Also assume that the video frame has already been converted into YCbCr format sub-sampled at 4:1:1 ratio, with each pixel value at each component represented as an 8-bit unsigned integer. SAD is used as a criterion to determine motion vector.
Assume that addition, subtraction, absolute value, comparison each takes one clock cycle to perform, and no two operations can be performed simultaneously. Ignoring memory access delay, compute the minimum clock frequency in GHz (rounded to 3rd fractional digit) that can support the motion estimation operation in real time.

(e) (7 points) Next, consider variable block size motion estimation as described in H.264 standard. One important observation is that the SAD value of a macro-block is the sum of all the SAD values of its sub-blocks. Implement a HLL (high level language) program that performs full search motion estimation of a single macro-block. The inputs of this program include a 16 × 16 current MB, and a 32 × 32 search region. The output of this program will be the 41 MVs and the corresponding SAD values. Create your own sample inputs (the 16 × 16 MB and 32 × 32 search region), run the program and compute the desired output. Submit (i) the source code of this program with sufficient comments; (ii) an explanation of the algorithm implemented in this program, especially how it make use of the observation mentioned above; (iii) printed output of the example you use. In addition. Post the program and the data you use on the homework web page AFTER the homework is due.