Assignment 2. Utilization of Shared Memory

Tridiagonal systems are a kind of linear equation systems where only the main diagonal, the diagonal above and the diagonal below contain non zero values. Tridiagonal systems are composed by a set of $N = 2^n$ linear equations with N unknowns

$$Au = d, \tag{1}$$

where A is a tridiagonal matrix $N \times N$ of the form

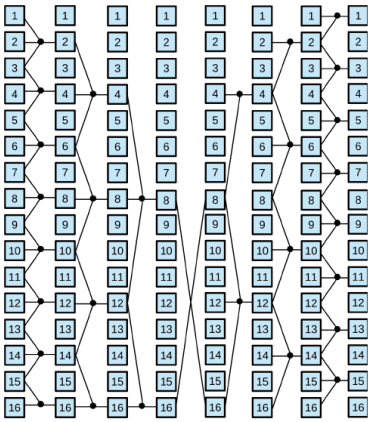
$$A = \begin{pmatrix} b_0 & c_0 & & & \\ a_1 & b_1 & c_1 & & \\ & a_2 & b_2 & c_2 & \\ & & \cdots & & \\ & & & \cdots & \\ & & a_{N-2} & b_{N-2} & c_{N-2} \\ & & & a_{N-1} & b_{N-1} \end{pmatrix} \tag{2}$$

There are several parallel algorithms for solving tridiagonal systems, but Cyclic Reduction (CR) is one of the most popular methods. CR comprises two phases, *forward reduction* and *backward substitution*. Forward reduction reduces a system to another with half the number of unknowns, until a 2-unknowns system is reached in $log_2 N - 1$ steps. Even-indexed equations are updated in parallel as linear combination of three equations deriving a system of only even-indexed unknown by:

$$a_i^{k+1} = -a_{i-1}^k s_1, \qquad b_i^{k+1} = b_i^k - c_{i-1}^k s_1 - a_{i+1}^k s_2, with \quad s_1 = \frac{a_i^k}{b_{i-1}^k}$$

$$c_i^{k+1} = -c_{i+1}^k s_2, \qquad d_i^{k+1} = d_i^k - d_{i-1}^k s_1 - d_{i+1}^k s_2, with \quad s_2 = \frac{c_i^k}{b_{i+1}^k}$$

where k denotes the step of algorithm. In each step of backward substitution, unknowns x_i are solved in parallel by substittuting the previously solved two x_{i-l} and x_{i+l} values to equation in n steps. The following figure show CR for a problem size $N = 16$:



1. Analyze and execute CR.cu. This code use a input diagonally dominant system which ensures numerical stability (Toeplitz matrix with row [-1 2 -1]), whose unknowns have the value **1.0 as solution**.

2. Write a version1 CR1.cu of this code in cuda (which you must submit) in such a way that the greatest parallelism can be extracted and using only a single invocation of a kernel. Arrays A, B, C and D have to be stored in Shared Memory during kernel execution. A, B, C and D are sent from the CPU to the GPU, and X is sent from the GPU to the CPU. The code must execute in batch form B systems of equations of size N, where each block executes one system. The number of systems of equations is $B = 2^{24}/N$.

3. Write a version2 CR2.cu of this code in cuda (which you must submit) with the same features as version 1 but each block can run multiple systems. In this case the statement of initialization of the execution parameters of dimBlock would be

$$dim3 \quad dimBlock(x, y, 1) \tag{3}$$

   where x would be the same value you used in version 1 when running a system of size N and y is the value of the number of systems running each block.

4. Complete the word A2-report file, with the requested data.