8/5/15

STEP1. Transfer matlab code to C

Problems:

1. Changeable size matrix causes array index out of bounds

A matrix in matlab can change size everywhere you want, and this kind of change is implicit. If declared in a small size in C, there will be no error reported, but the result is wrong.

In C these matrix should be declared in their maximum size.

Usually there will be a red wave under the matrix (showing a warning), this is a sign that this matrix should not be declared in its original size.

1. Index mismatch

In matlab, matrix indices start from 1, in C from 0. Indices must be transferred VERY CAREFULLY, especially when array entries to be indices. In matlab 1:n transferred to for (int i = 0; I < n ; i++ ) in C.

1. Global and local variables

In Matlab variables can be declared anywhere you need, and by default they are global. Such variables, which are used in more than one module, must be declared global, and static if necessary. Notice that if the global variable is used in a loop, they should be initialized explicitly every time before entering loop.

1. Matlab Coder

if the indices are not continuous, C code cannot be generated. The indices should be stored in a vector so that the indices of the vector are continuous to access.

STEP2. HLS.

Design step:

Simulation,systhesis,

* If the loop iteration limit is a variable, Vivado HLS cannot determine the maximum upper limit.
* The TRIPCOUNT directive can applied to the loop to manually specify the number of loop iterations and ensure the report contains useful numbers.

Concurrent hardware

Advantages:

Parallel operation. Re-order the sequence, finished in fewer clock.

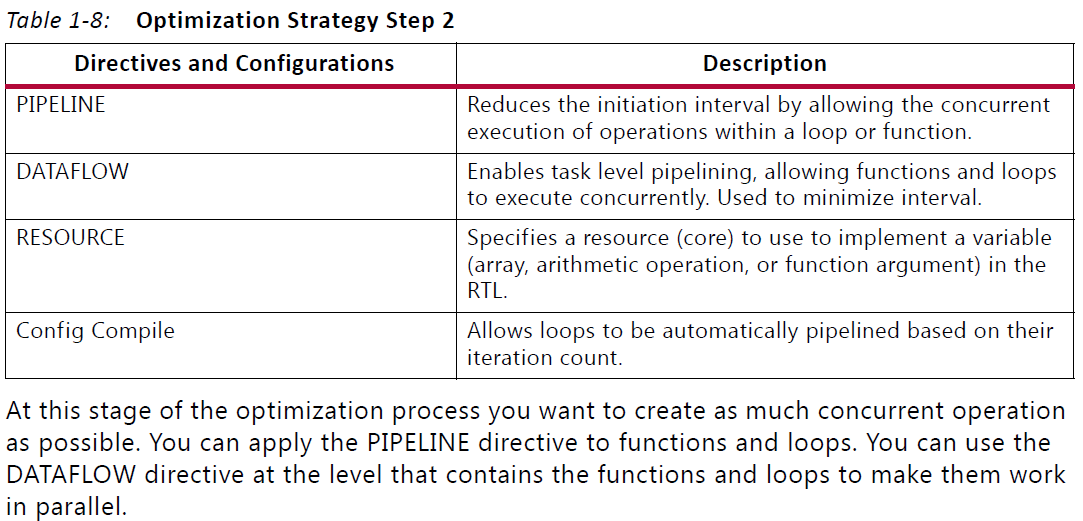
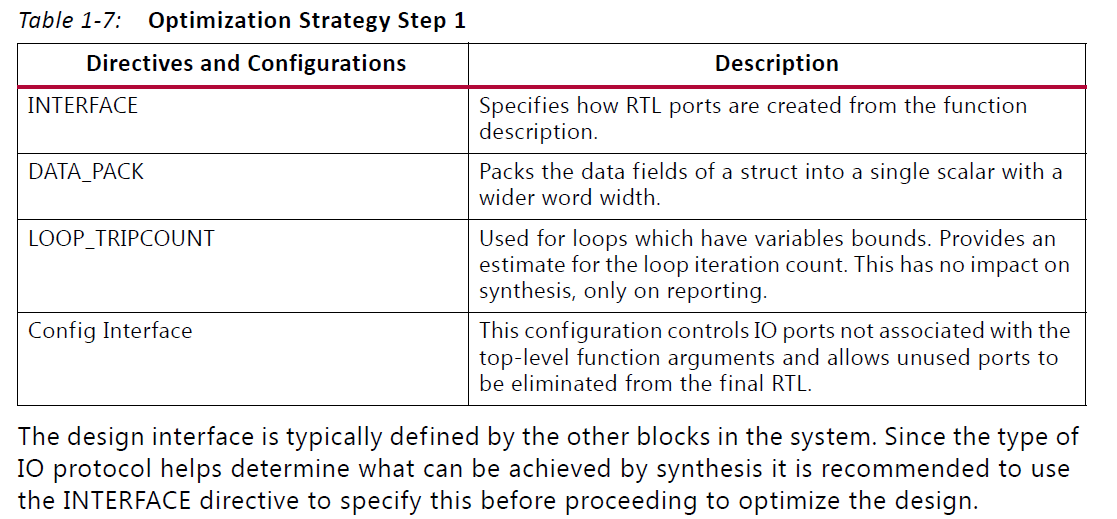
Pipelined operation. Using all resources in the same time.

Pipelined tasks: DATAFLOW can be used so that functions/loops are executed as soon as the data is available.

Systhesis of loops. Loops are scheduled in the same order they appear in the code, even if there is no dependency between each loop.

Systhesis of arrays. Usually in BRAM. Can be partitioned or packed.

General steps for optimization:



**Strategies for pipelining:**

PIPELINE:

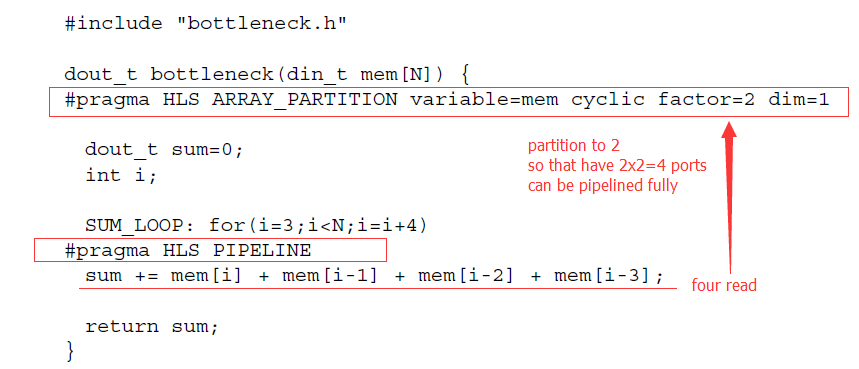
Sub-function is not automatically pipelined.

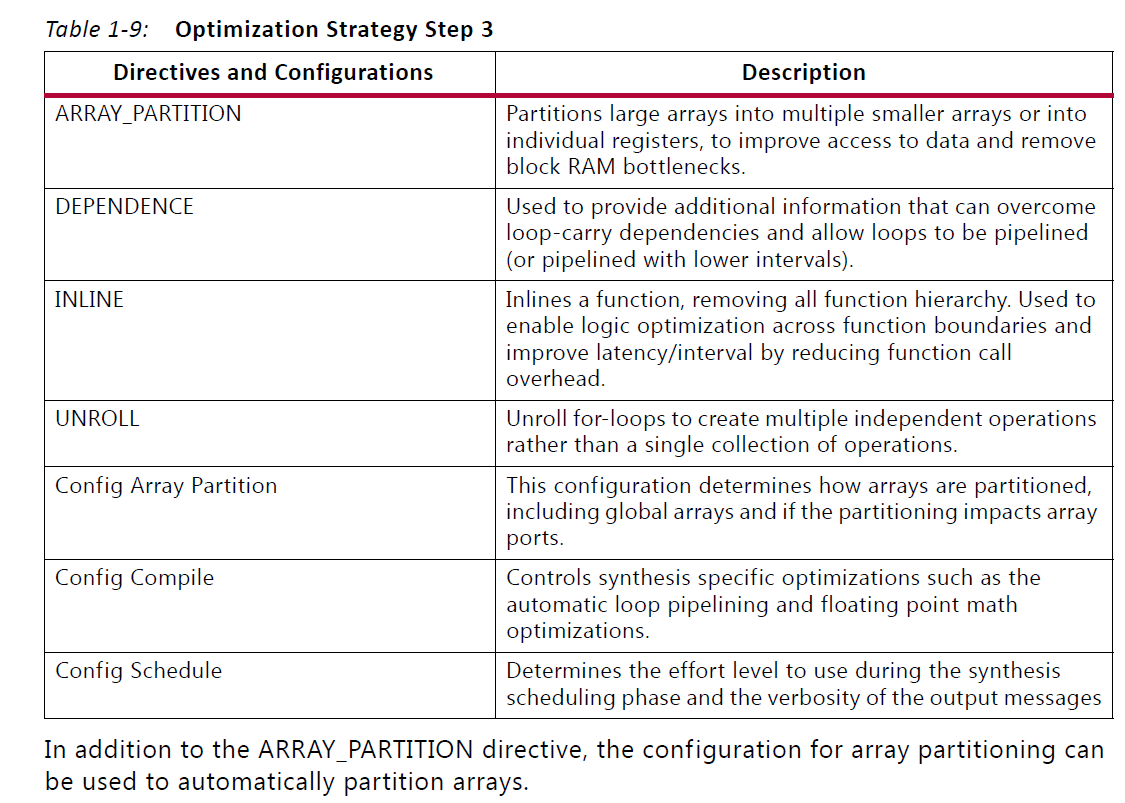
Sub-loop is automatically pipelined.

Loops with variable bounds cannot be unrolled. So does its above hierarchical loops and functions. So pipeline it and use DATAFLOW to maximize the performance.

**Function with a Top-Level Loop Processing Blocks**

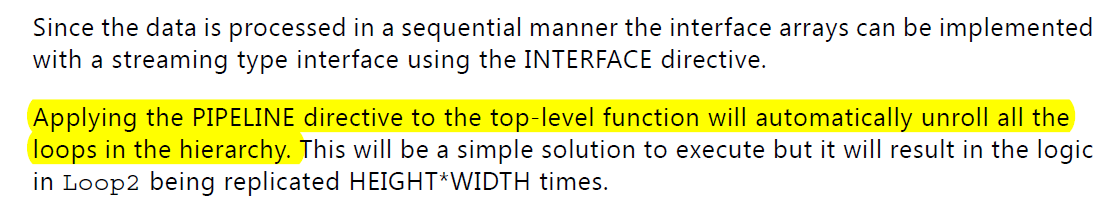
* Only two data ports in BRAM so maximum two reads are allowed at same cycle. So cannot be pipelined fully. Use ARRAY\_PARTITION to split into small arrays so that can have more data ports.



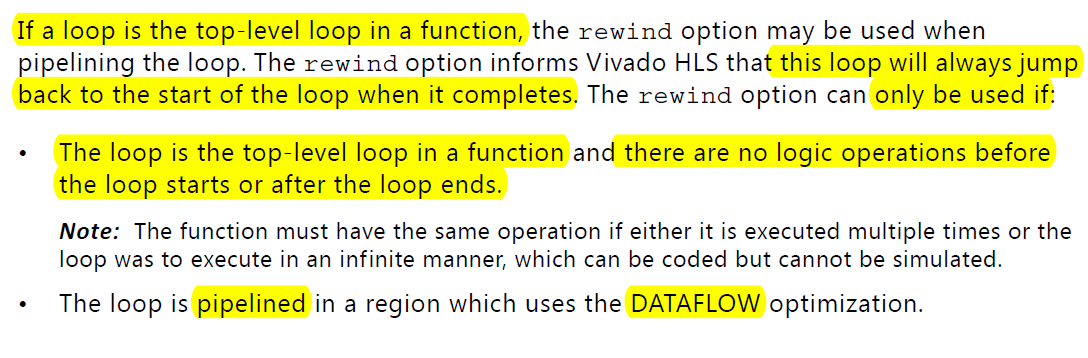


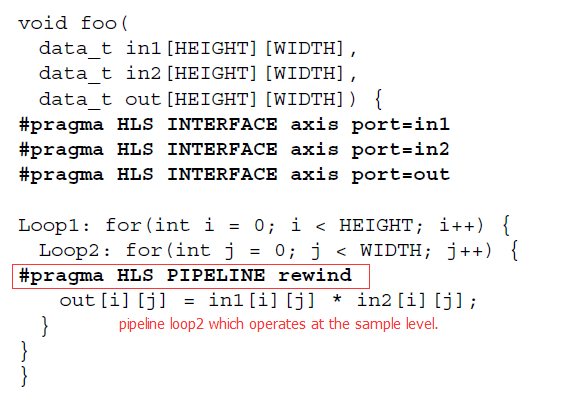
* If the code contains several loops with similar indexing, MERGE\_LOOP to merging the loop.
* The key strategy dealing with regions processing a block of data, is pipelining the loops.
* The most optimum loop to pipeline: the lowest level of loop which operates on a data sample.
* When nested loops are pipelined the LOOP\_FLATTEN optimization is automatically applied to the nested loops. This automatically removes any additional cycle between Loop1 and Loop2.
* **Pipelining top level function**

PIPELINE



REWIND

Doing so results in a design which processes one sample per clock and immediately starts to execute the next set of samples, without pause.

Example: 

**Function with Nested Operations**

Error 2 error C4996: 'fopen': This function or variable may be unsafe. Consider using fopen\_s instead. To disable deprecation, use \_CRT\_SECURE\_NO\_WARNINGS. See online help for details. c:\users\admin\documents\visual studio 2013\projects\femv8\femv8\fem\_tb.c 12 1 FEMv8