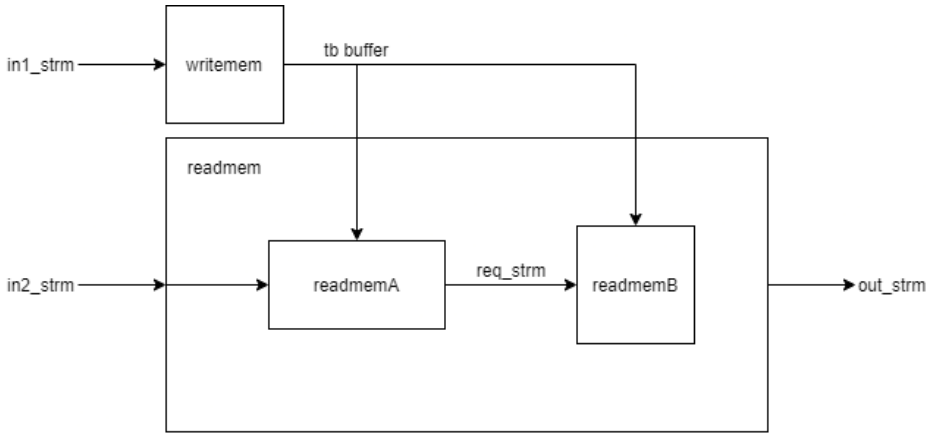
Dataflow Stable Content

From Xilinx example Design

1. Introduction

There are four functions in the system. The “writemem” function reads from a stream and fills up the tb buffer. The “readmem” function is designed to implement DATAFLOW between “readmemA” and “readmemB”. “readmemA” reads a stream of indexes to tb and writes out the lookup as a stream “req_strm”. On the other hand, “readmemB” reads the previous stream, “req_strm”, as indexes to another tb lookup and writes the sum of the looked up numbers. The problem we are facing now is that the tb is read from both sub-functions, which will make the DATAFLOW fail.

2. System Diagram



The two inputs, in1_strm and in2_strm, are input signal for the function “writemem” and “readmem” respectively. In “writemem”, data in in1_strm is stored in the tb buffer for “readmem”. The function “readmem” will produce out_strm with two inputs, tb and in2_strm. To optimize the system, we split readmem into two subfunctions, “readmemA” and “readmemB”. The output of “readmemA” is reg_strm, which will be sent into “readmemB” to produce the output signal out_strm. We may use the dataflow technique on these two subfunctions. The loops in each function have the PIPELINE pragma as well.

However, since tb is read from both sub-functions, DATAFLOW won't work unless we use the HLS stable pragma. The stable pragma marks variables within a DATAFLOW region as being stable. It applies to both scalar and array variables whose content can be either written or read by the process inside the DATAFLOW region. This pragma eliminates the extra synchronization involved for the DATAFLOW region. Therefore we can solve the problem with this pragma.

3. Comparison

Let's compare the synthesis result between baseline solution and optimized solution.

		baseline	Optimaized
Latency (cycles)	min	54	35
	max	54	35
Latency (absolute)	min	0.720 us	0.467 us
	max	0.720 us	0.467 us
Interval (cycles)	min	54	35
	max	54	35

	baseline	Optimaized
BRAM_18K	1	2
DSP48E	0	0
FF	99	476
LUT	401	1305
URAM	0	0

The latency is greatly improved with the dataflow and pipeline technique. On the other hand, since the loops are unrolled, the usage of hardware will, of course, increase a certain degree. It is worth noting that the synthesis cannot be done if we remove the stable pragma. The error messages are shown as below.