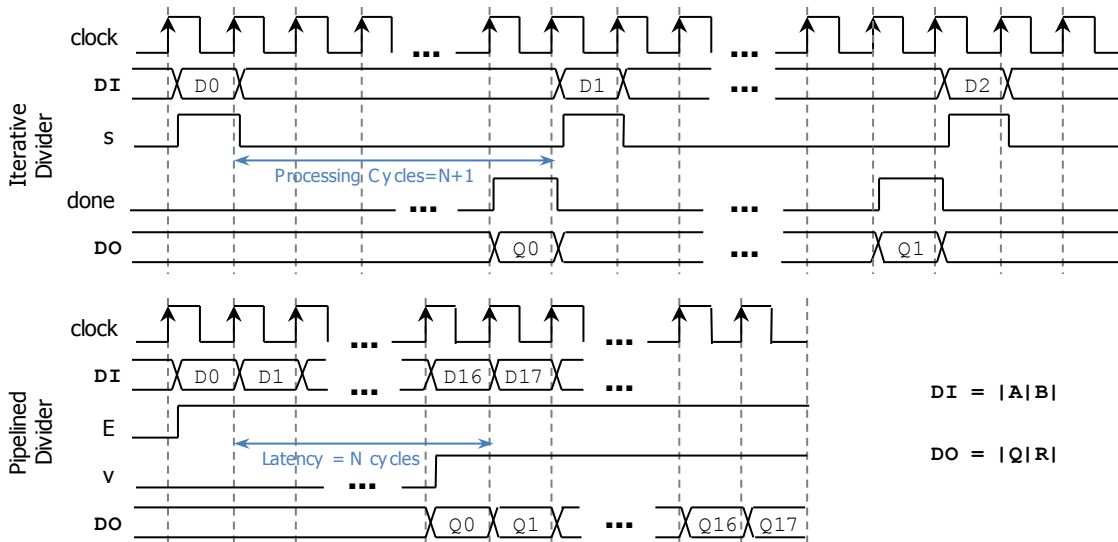


# Homework 3

(Due date: June 9<sup>th</sup>)

## PROBLEM 1 (20 PTS)

- Performance Analysis: Iterative Integer Divider vs. Pipelined Integer Divider (N=M=16):**
  - Iterative Divider Operation:** Input data (16-bit A, 16-bit B) is read when the *s* signal (a one-cycle pulse) is asserted. After  $N+1=17$  cycles, the result (16-bit Q, 16-bit R) is ready with *done*=1. Only after this, we can feed new data. To process data as fast as possible, we must issue *s*=1 (with new data) right after *done*=1.
  - Pipelined Divider Operation:** The circuit reads input data (16-bit A, 16-bit B) when the enable (*E*) signal is asserted. After a processing delay of  $N=16$  cycles, the result (16-bit Q, 16-bit R) is ready and it is signaled by *v*=1. Unlike the iterative divider, we can continuously feed data (with *E*=1). To process data as fast as possible, we must keep *E*=1 (with new data) every clock cycle.



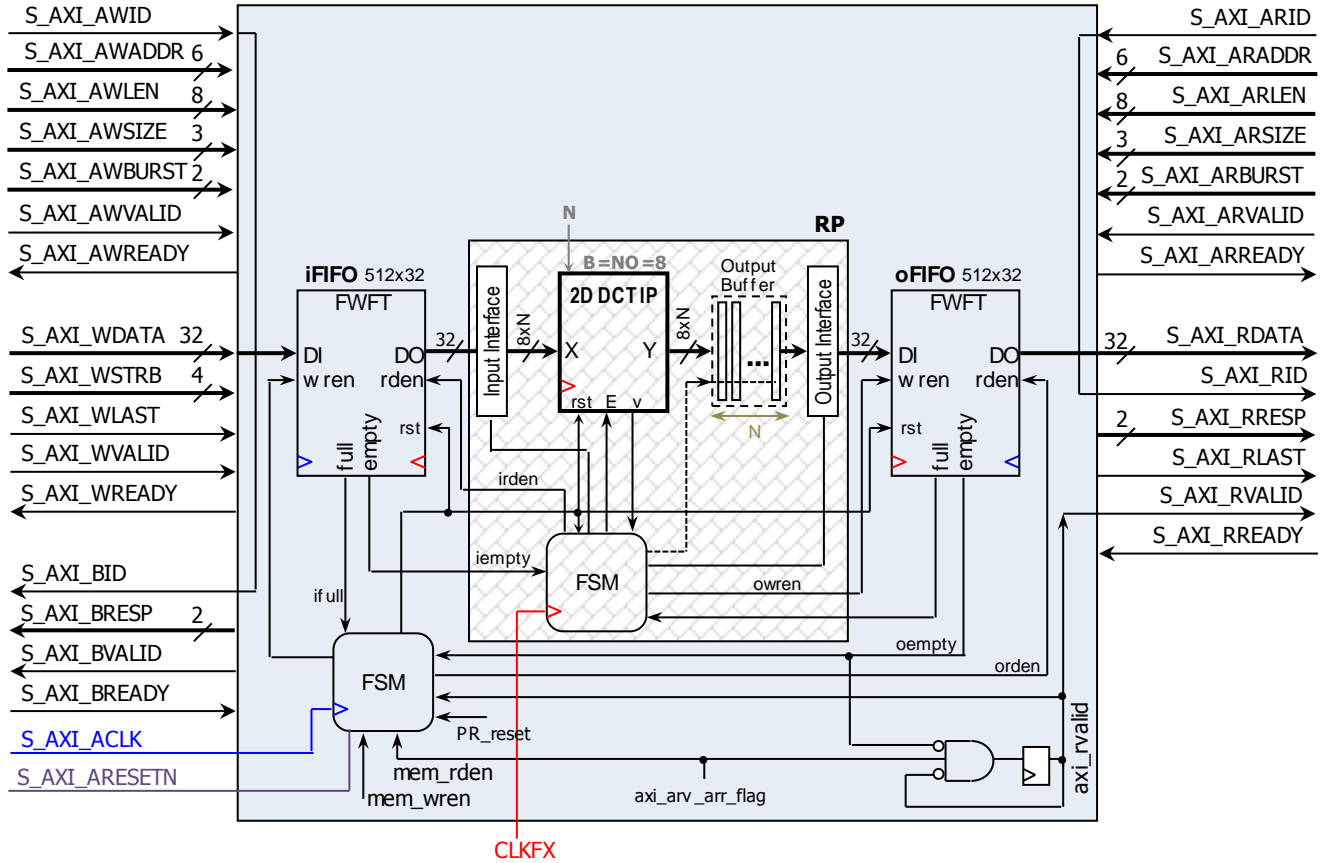
- An operation is defined as the computation of one input data set. The processing cycles for *P* operations is given by:
  - Iterative Divider:** It can compute *P* operations in  $P \times (N+2)$  cycles (1 operation is processed in  $N+1$  cycles, but there is a one cycle delay before we can start the next operation)
  - Pipelined Divider:** It can compute *P* operations in  $N + (P-1)$  cycles.
- In the following table, complete the number of processing cycles, processing times (us), and operations per second.
  - Use  $T_{\text{clock}} = 8 \text{ ns}$  (same as the  $PL\_CLK = 125 \text{ MHz}$  input clock in ZYBO or ZYBO Z7-10)
  - The metric Operations per second is an average based on a given number of operations. Example: if a circuit can process 20 operations in 1 us, then we have  $\frac{20 \text{ operations}}{1 \text{ us}} = 20 \times 10^6$  operations per second.

P	Iterative Divider			Pipelined Divider		
	Processing cycles	Processing Time (us)	Operations per second	Processing Cycles	Processing Time (us)	Operations per second
100						
1000						
10000						
100000						

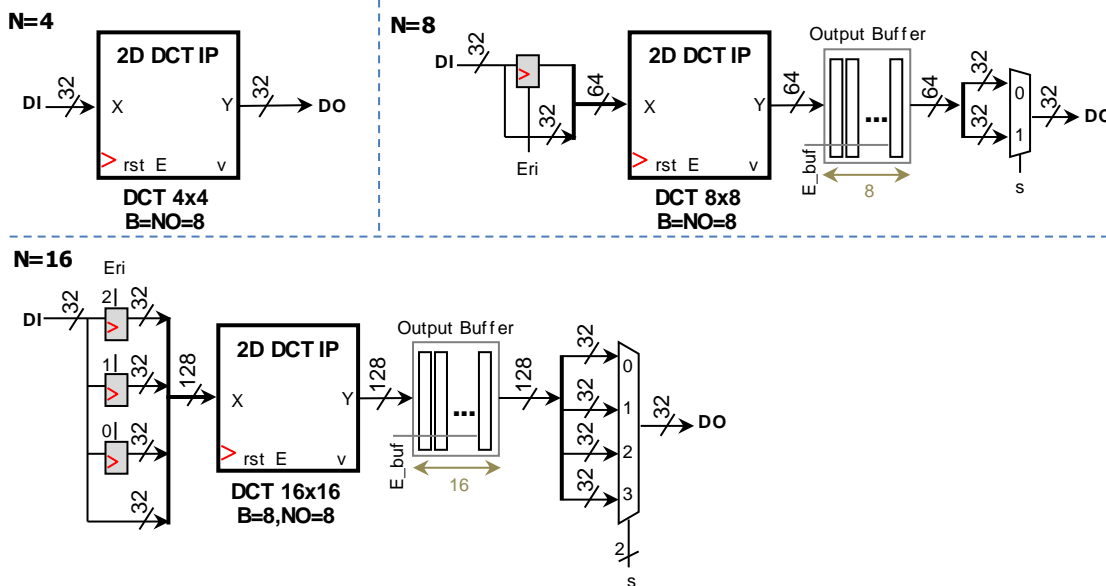
- For the Iterative Divider: Is the Operations per second constant? Yes or No? Why?
- For the Pipelined Divider: If  $P \rightarrow \infty$ :
  - How many operations are computed per cycle?
  - What is the Operations per second?

PROBLEM 2 (15 PTS)

- The figure shows the 2D DCT IP AXI4-Full Peripheral. It includes a Reconfigurable Partition (RP). For this particular PR implementation, we allow for N to be run-time reconfigurable (N=4,8,16), while we fix the parameters B=NO=8.



- The input and output of the 2D DCT IP require more than 32 bits when N = 8, 16. This requires an Input interface to the iFIFO and an Output interface to the oFIFO. The figure shows the different interfaces for each N (4, 8, 16) when B=NO=8. As the FSM @ CLK\_FX controls data flow from the input and the output, it depends on N.



- We want to build a dynamically reconfigurable system, where we can change N (4, 8, 16) at run-time:
  - ✓ The RP (Reconfigurable Partition) is depicted in the figure. The Output Buffer, the Input interface and the output interface to FIFOs, as well as the FSM @ CLK\_FX are included in the RP. Why is this necessary?
  - ✓ Signal *rst*: Active-high signal generated by the FSM @ S\_AXI\_ACLK. It resets the 2D DCT IP, the red FSM, and the FIFOs. Why is this signal important? Do we assert this signal before or after performing DPR? Why?
  - ✓ The RP outputs toggle during DPR. What could happen to the contents of oFIFO during DPR?

### PROBLEM 3 (65 PTS)

- Attach your Project Status Report (no more than 1 page, single-spaced, 2 columns, only one submission per group). This report should contain the current status of your project. For formatting, use the provided template (Final Project - Report Template.docx). The sections included in the template are the ones required in your Final Report. At this stage, you are only required to:
  - ✓ Include a project description.
  - ✓ Specify a (tentative) allocation of tasks in: i) software routine, and ii) reconfigurable hardware.
    - If you plan to use run-time alterable hardware, indicate what tasks it will be doing.
  - ✓ Hardware Architecture: Include a Draft Block Diagram with (tentative) I/O description and I/O mechanism.
- As a guideline, a generic hardware/software partitioning of an application is depicted. The figure shows the tasks performed by the software routine and the PS peripherals we plan to use. It also shows a Block Diagram of the Hardware with generic I/Os. The Reconfigurable Partition (RP) is also depicted. Note that this hardware uses external I/Os to the PL.

