PARAMETERIZED DCT DA IMPLEMENTATION

This is the hardware implementation of the DCT using Distributed Arithmetic.
Data sequence: $x_n,\ n = 0,1,\dots,N-1.$ N-point DCT: $z_k,\ k = 0,1,\dots,N-1$:

$$z_k = \frac{2}{N}c(k)\sum_{n=0}^{N-1} x_n cos\left(\frac{\pi(2n+1)k}{2N}\right), c(0) = \frac{1}{\sqrt{2}}, c(k) = 1, k = 1,\dots,N-1$$

I.1 REGULAR DCT (1D) WITH NO DECOMPOSITION:

Without loss of generality, let's omit $^2/_N$. Then, the DCT is given by: $\bar{z} = T(N)\bar{x}$

where: $\bar{z} = [z_0 \quad z_1 \quad \cdots \quad z_{N-1}]^T$, and $\bar{x} = [x_0 \quad x_1 \quad \cdots \quad x_{N-1}]^T$

$T(N)$: $N \times N$ matrix whose $(k,n)th$ component is: $T(N)_{(k,n)} = c(k)cos\left(\frac{\pi(2n+1)k}{2N}\right)$

- Here, we require N inner products, each of size N.

I.1. REGULAR DCT (1D) WITH EVEN-ODD DECOMPOSITION

If N is even, it has been shown that the DCT can be computed in a recursive fashion with the even-odd decomposition (1st iteration of the recursive DCT algorithm). Thus, $\bar{z} = T(N)\bar{x}$, turns into:

$$\begin{bmatrix}\bar{z_e}\\ \bar{z_o}\end{bmatrix} = \begin{bmatrix}T(N/2) & T(N/2)\\ D(N/2) & -D(N/2)\end{bmatrix}\begin{bmatrix}\overline{x_p}\\ \bar{I}\overline{x_r}\end{bmatrix} = \begin{bmatrix}T(N/2)\left(\overline{x_p} + \bar{I}\overline{x_r}\right)\\ D(N/2)\left(\overline{x_p} - \bar{I}\overline{x_r}\right)\end{bmatrix} \quad (1)$$

Where:

$$\bar{z_e} = [z_0 \quad z_2 \quad \cdots \quad z_{N-2}]^T, \quad \bar{z_o} = [z_1 \quad z_3 \quad \cdots \quad z_{N-1}]^T$$
$$\overline{x_p} = [x_0 \quad x_1 \quad \cdots \quad x_{N/2-1}]^T, \quad \overline{x_r} = [x_{N/2} \quad x_{N/2+1} \quad \cdots \quad x_{N-1}]^T$$
$$\bar{I} = \begin{bmatrix}0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\\ \vdots & \iddots & \vdots & \vdots\\ 1 & 0 & 0 & 0\end{bmatrix}, D\left(\frac{N}{2}\right) = LST\left(\frac{N}{2}\right)Q$$

- For the definitions of L,S,Q (and other details), see:
  S. Yu, E. Swartzlander, "DCT Implementation with Distributed Arithmetic", *IEEE Transactions on Computers*, vol. 50, no. 9, Sep. 2001.
- For our hardware implementation, $T(N/2), D(N/2)$ are matrices whose values are provided.
- Note that $\bar{z_e}$ can be further decomposed (more iterations of recursive DCT algorithm). However, $\bar{z_o}$ cannot be further decomposed. Thus, it is strongly advised that N be a power of 2.

Even-odd decomposition: with N even, let's call: inner summations/subtractions as w, and $M = N/2$.

'w' with even indices. M summations	'w' with odd indices. M subtractions
$w[0] \leftarrow x[0] + x[N-1]$	$w[1] \leftarrow x[0] - x[N-1]$
$w[2] \leftarrow x[1] + x[N-2]$	$w[3] \leftarrow x[1] - x[N-2]$
$\vdots$	$\vdots$
$w[i_e] \leftarrow x[i_e/2\ ] + x[N-(i_e/2+1)]$	$w[i_o\ ] \leftarrow x[\lfloor i_o/2 \rfloor] - x[N-(\lfloor i_o/2 \rfloor+1)]$
$\vdots$	$\vdots$
$w[N-2] \leftarrow x[N/2-1] + x[N/2]$	$w[N-1] \leftarrow x[N/2-1] - x[N/2]$

* $i_o = i_e + 1$

- We thus require N inner products, each of size $N/2$.

Example: To understand this better, let's see an example with N=16. Eq. (1) would look like:

$$\begin{bmatrix}z[0]\\ z[2]\\ z[4]\\ z[6]\\ z[8]\\ z[10]\\ z[12]\\ z[14]\end{bmatrix} = \begin{bmatrix}\end{bmatrix}\begin{bmatrix}x[0]+x[15]\\ x[1]+x[14]\\ x[2]+x[13]\\ x[3]+x[12]\\ x[4]+x[11]\\ x[5]+x[10]\\ x[6]+x[9]\\ x[7]+x[8]\end{bmatrix}, \begin{bmatrix}z[1]\\ z[3]\\ z[5]\\ z[7]\\ z[9]\\ z[11]\\ z[13]\\ z[15]\end{bmatrix} = \begin{bmatrix}\end{bmatrix}\begin{bmatrix}x[0]-x[15]\\ x[1]-x[14]\\ x[2]-x[13]\\ x[3]-x[12]\\ x[4]-x[11]\\ x[5]-x[10]\\ x[6]-x[9]\\ x[7]-x[8]\end{bmatrix}$$

This requires 16 inner products, each of size 8.

II. HARDWARE IMPLEMENTATION

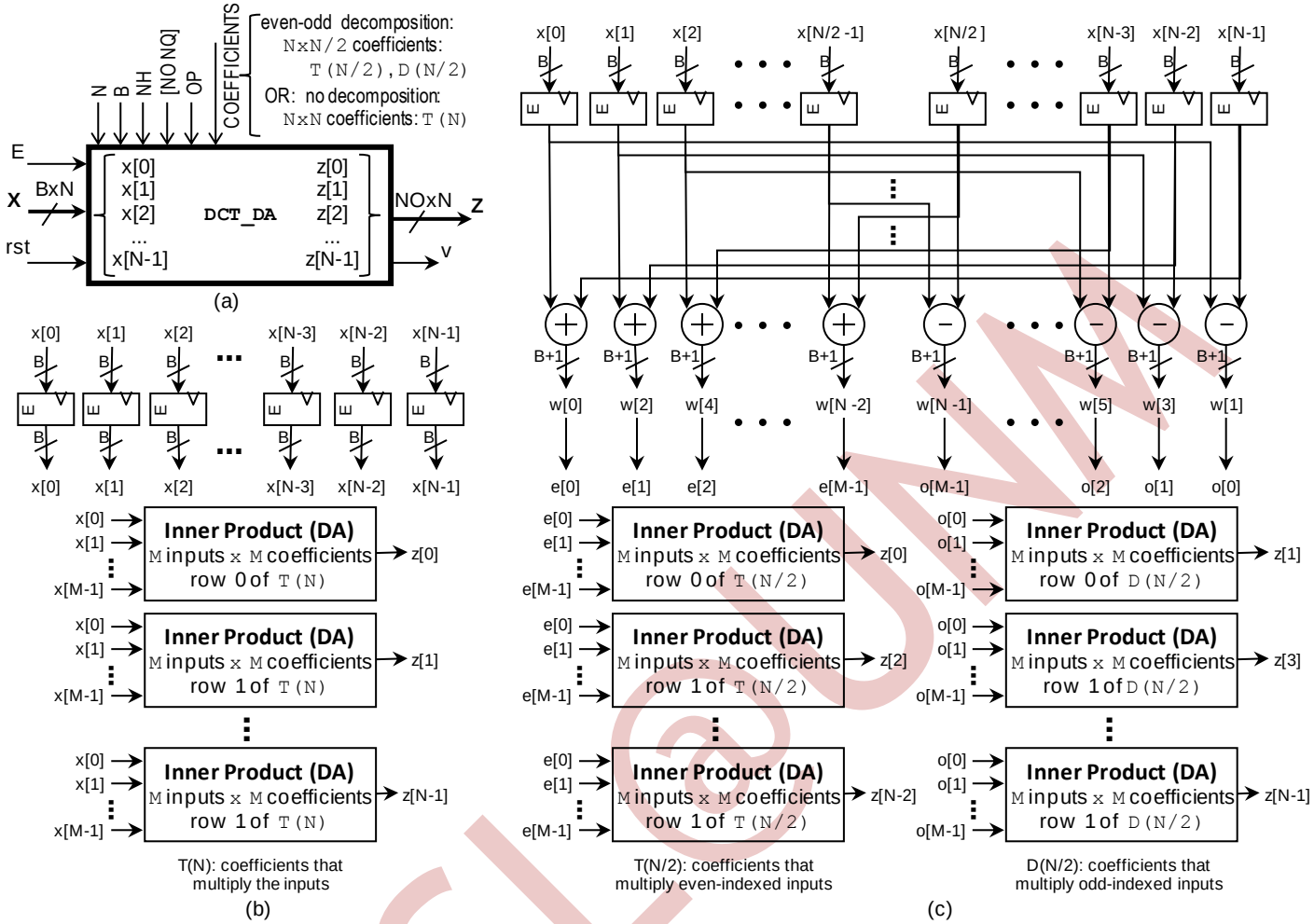

Figure 1. Parameterized DCT DA (1D). (a) IP core. Even-odd decomposition: N=8,16, M=N/2. No even-odd decomposition: N=2,3,4,5,6,7, M=N. (b) Inner architecture of DCT_DA (1D) with no even-odd decomposition. N inner products, each of N products (c) Inner architecture of DCT_DA (1D) with even-odd decomposition. N inner products, each of N/2 products.

Fig. 1(a) depicts a parameterized 1-D DCT-DA IP core that allows modification of parameters such as the DCT size, DCT coefficients, and the bit precision. The DCT-DA IP core is shown with its inputs, outputs, and parameters. Signal 'E' controls the input validity. The following is a description of the parameters:

- N: DCT size.
- NH: Number of bits of the coefficient values.
- B: Number of bits of the input values.
- L: LUT input size. We can choose the input bit-width of the internal LUTs which are the ones that contain the coefficient values' information. $L \geq 2$ (otherwise there is no LUT)
- OP: Output truncation scheme:
  - $OP = 0$: A truncation is performed at the LSBs, and then saturation is performed, so that the output fixed-point format results $[NO\ NQ]$.
  - $OP = 1$: A truncation is performed at both the LSB and MSB parts, so that the output fixed-point format results $[NO\ NQ]$.
  - $OP = 2$: The maximum number of allowable bits is used for the output format, which is given by the following formula: $[NH + B - 2 + \lceil \log_2(N+1) \rceil - 1 \quad NH + B - 2]$
- $[NO\ NQ]$: Output fixed-point format (valid if $OP \neq 2$). NO: number of integer bits. NQ: number of fractional bits. The user can choose the output format, provided is within the bounds of the maximum allowable number of bits for both the integer and fractional part.
- COEFFICIENTS: $T(N/2), D(N/2)$ (even-odd decomposition). $T(N)$ (no even-odd decomposition). A text file with the LUT values that represent those matrix is read by the DCT-DA VHDL code.

II.1. NO EVEN-ODD DECOMPOSITION: One set of coefficients: $T(N)$: Each row is not symmetric. $N = M$. M multiple of L. Fig. 1(b) depicts the architecture. $sizeI = B$: The size of the signals that enter the inner product is B. Each inner product is the same as the one in FIR_DA. This hardware requires many inner products in parallel. If N=16, we need 16 inner products of 16 non-symmetric coefficients. Fig. 2(a) shows the detail of the inner products with $T(N)$.

II. 2. EVEN-ODD DECOMPOSITION: Two sets of coefficients: $T(N/2)$ and $D(N/2)$. These matrices are not symmetric. Each row is not symmetric. N is even, $M = N/2$, and M is multiple of L. Fig. 1(c) depicts the architecture. $sizeI = B$: The size of the signals entering the inner product is $B + 1$. Each inner product is similar to the one in FIR_DA (the difference is the arrangement of the input signals). This hardware require many inner products in parallel. If N=16, we need 16 inner products of 8 non-symmetric coefficients.
Note that we rename $w[k], k = 0, \dots, N-1]$, by $e[j], o[j], j = 0, \dots, M-1$ for a clearer explanation. Fig. 2(b) shows the detail of the inner product core for both the inner products with $T(N/2)$ and $D(N/2)$.

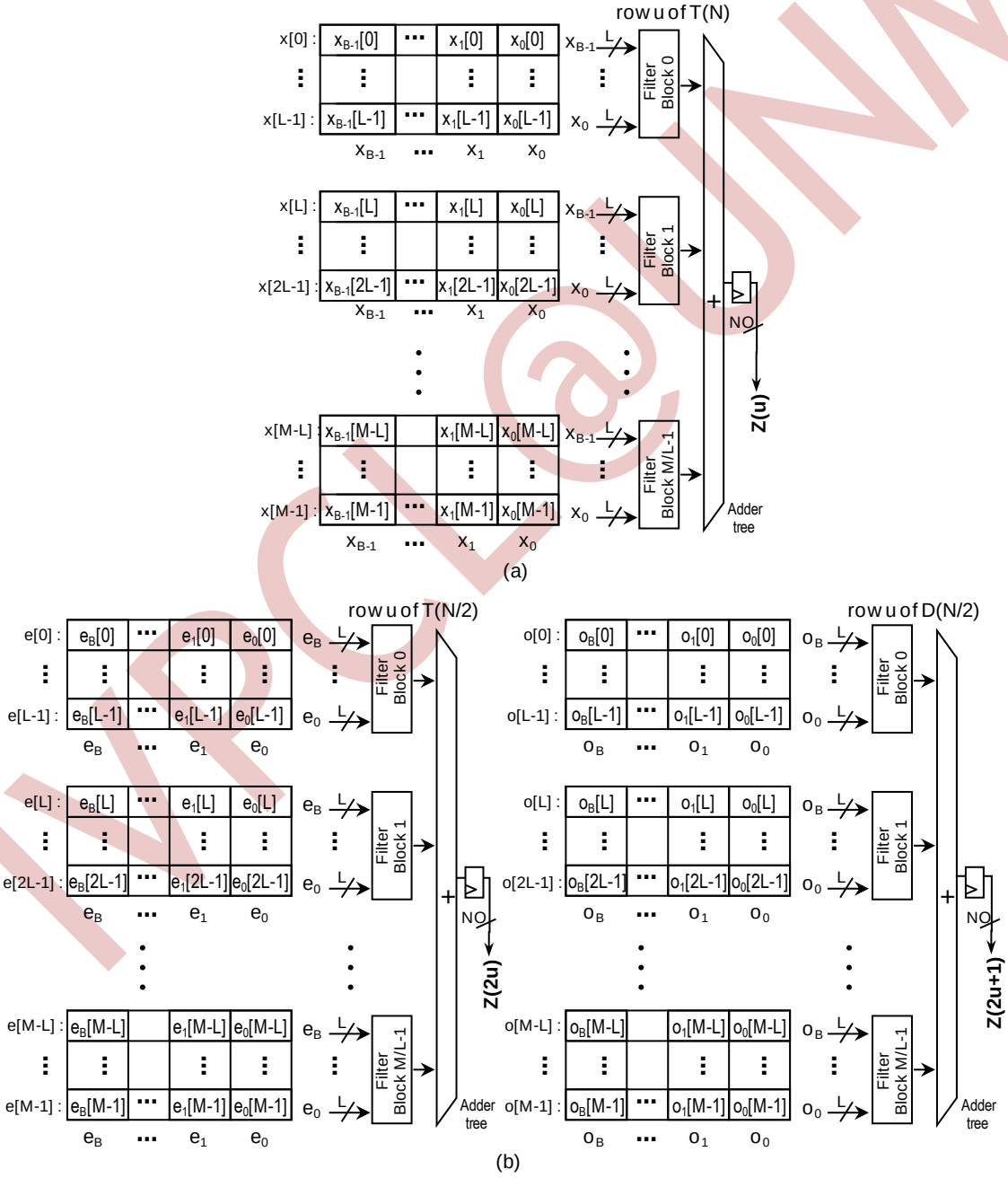

Figure 2. Inner product details. The filter blocks are exactly the same used in FIR_DA. (a) No even-odd decomposition, M=N (b) Even-odd decomposition. Inner product details for both even and odd parts. M=N/2, N even. M is multiple of L.

II.3. HARDWARE FORMULAS (for both even-odd decomposition and no-decomposition)

- The output largest number of bits of each inner product can be calculated with the inputs of B bits and the coefficients of NH bits. Also, without loss of generality, we assume that the inner product is always of size N. In this case, the derivation is identical to that of a filter with N coefficients. Thus, the largest number of output bits is given by:

$$L\_DCT = NH + B + \lceil \log_2(N+1) \rceil - 1$$

- As for the maximum word length for the filter block output, this can be calculated in an identical way as in the filter block output of the FIR_DA. The input has $sizeI$ bits, the coefficients have NH bits, and the number of products is L. Then, for the filter block output, it results:

$$L\_fbk = NH + sizeI + \lceil \log_2(L+1) \rceil - 1$$

- Finally, if the input X has the format $[B \;\; B-1]$, and the coefficients have the format $[NH \; NHB-1]$, then the output largest format will have $NH + B - 2$ fractional bits. Then, the largest output format results:

$$[L\_DCT \quad NH + B - 2]$$

- I/O delay: $REG_{LEVELS} = \lceil \log_2(sizeI) \rceil + \lceil \log_2(M/L) \rceil + 2$

IMPORTANT NOTES ABOUT VHDL PROJECT DCT_v2 and DCT2d_V2:

The project comes with the text files containing the coefficients for the following cases:

- Even-odd decomposition: N=8,16, NH=10,12,14,16
- No decomposition: N=2,3,4,5,6,7, NH=10,12,14, 16

So, the COEFFICIENTS parameter has been already taken care of.

III. 2D DCT IMPLEMENTATION:

The 2D- DCT IP features some extra parameters: $[NOa\ \ NQa]$ and TR. The parameter TR is very important as it lets us pick the number of transpose memories:

- $TR = 1$: only one transpose memory exists and the process requires more cycles.
- $TR = 2$: two transpose memories, resulting in a fully pipelined system.

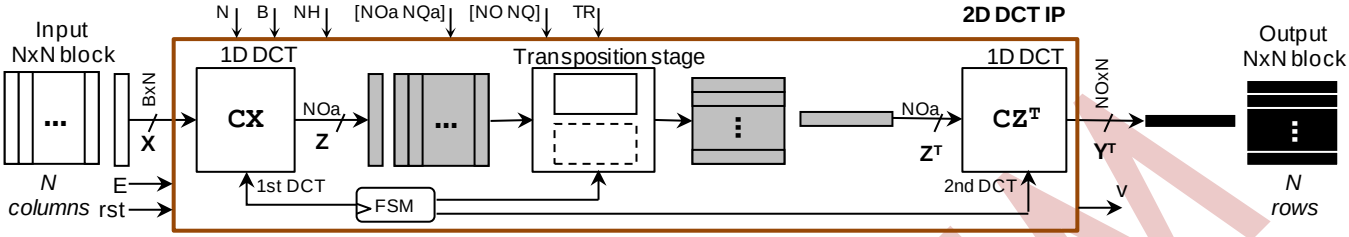

Figure 3. 2D DCT implementation. TR=1: one transpose memory, TR=2: two transpose memories (ping-pong mode or fully-pipelined mode).

IV. PLB PERIPHERAL:

Figure 4 shows a PLB interface for the 2D DCT IP. Note that we require an output buffer to hold data that usually appears at a faster rate than what the oFIFO can retrieve.

$NWIC = \frac{N}{PLBW/B}$, # of words needed for an input column

$NWOC = NWIC \times \left\lceil \frac{NO}{B} \right\rceil$, # of words needed for an output column

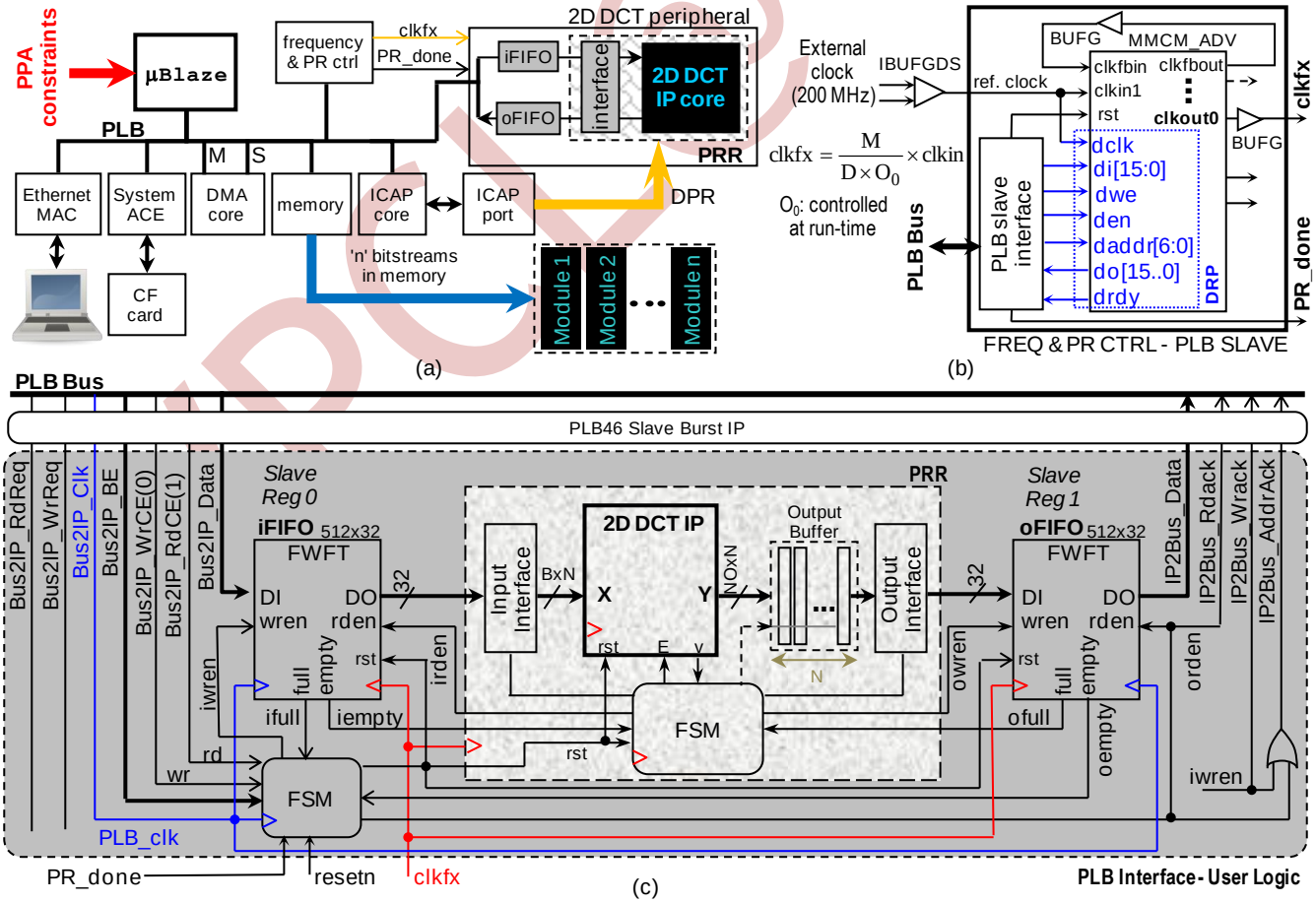

Figure 4. PLB interface for 2D DCT

The following table displays the values of NWIC and NWOC for different DCT sizes and values of B and NO:

DCT	B=8	NO=8	NO=16
4x4	NWIC = 1	NWOC = 1	NWOC = 2
8x8	NWIC = 2	NWOC = 2	NWOC = 4
16x16	NWIC = 4	NWOC = 4	NWOC = 8

For our purposes, we will use B = 8, and NO=8,16:

- NO = 16: for every input pixel, twice the amount of data is generated as output. NO=16 covers the cases NO = 10,12,14.
- If B = 8 and NO = 8 → NWOC = NWIC. If B = 8 and NO = 16 → NWOC = 2*NWIC

IV.1. Input/Output Interface:

Figure 5 depicts the different I/O interfaces to the 2D DCT IP core. Note that the iFIFO outputs 32-bit words and the oFIFO gets 32-bit words. Thus, whenever we have input/output data larger than 32 bits, we require some extra logic in the input and/or output.
Note that when DCT=4x4 and B=NO=8, there is no need for an output buffer. In all other cases, we do need an output buffer as the oFIFO is 32-bit wide. Actually, in this case, there is no need for an I/O interface at all.

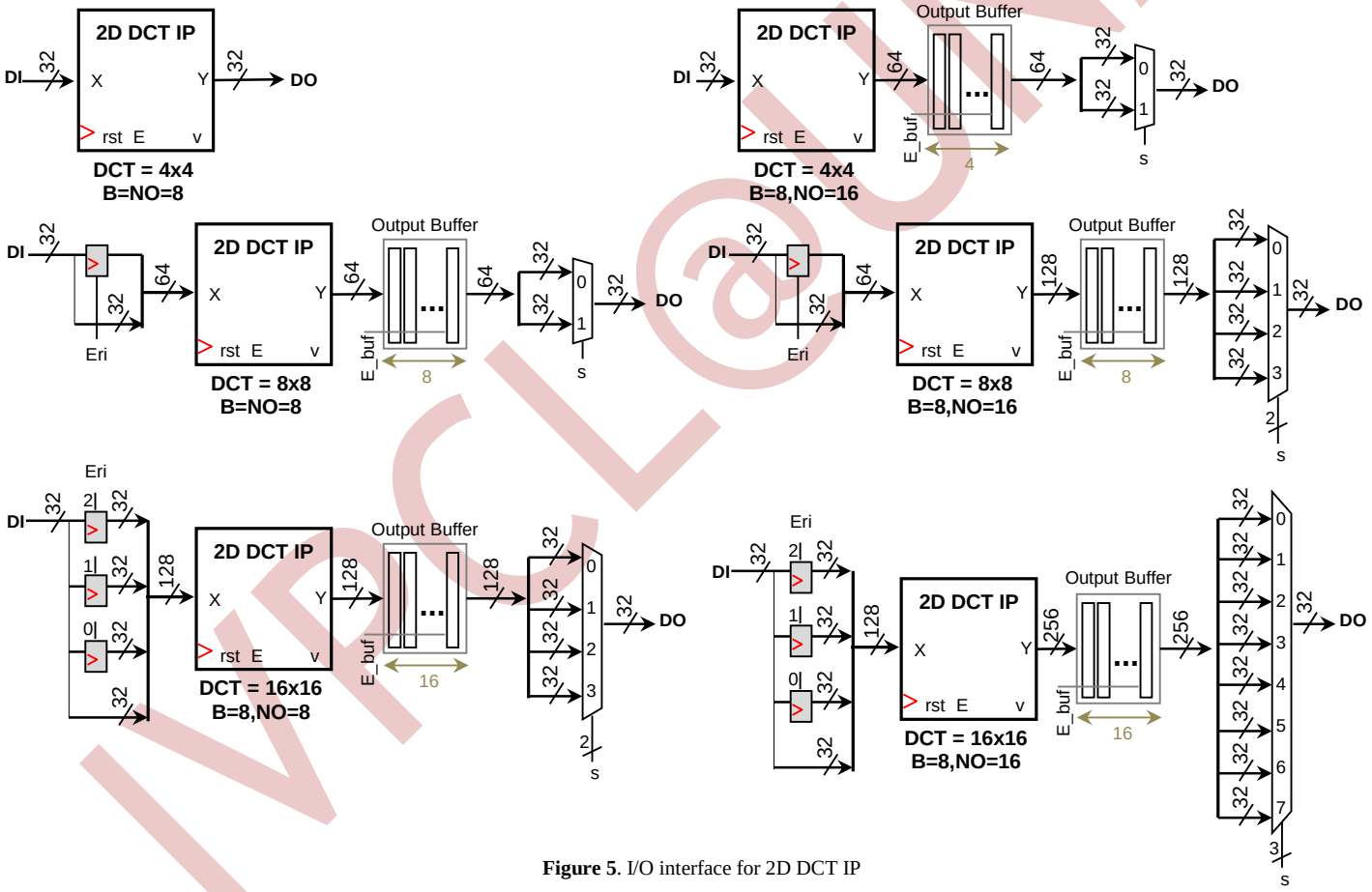

Figure 5. I/O interface for 2D DCT IP

Figure 6 depicts the timing diagram of two instances of the input interface behavior: DCT = 8x8 and DCT=16x16. Note how Eri, $irden$, and E need to be generated whenever a new input data is available at the output of the iFIFO.

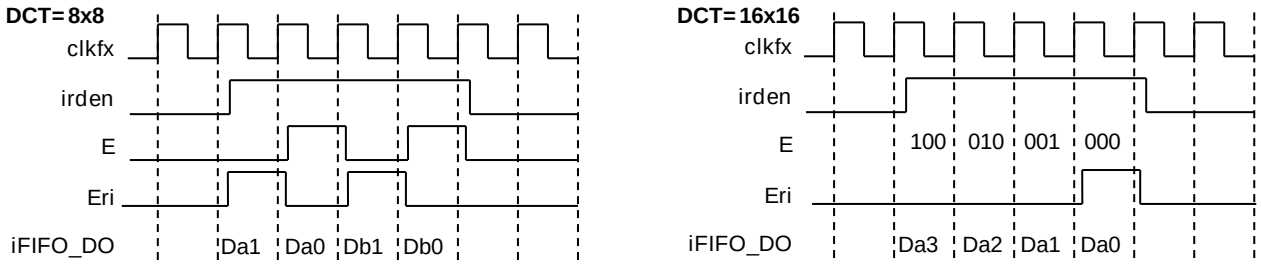

Figure 6. Timing diagram for DCT=8x8 (left) and DCT=16x16 (right)

IV.1.1. Output buffer:

The output rows will be generated consecutively (one cycle after the other) at the output ALWAYS, even if we didn't feed the data in consecutively. In most cases (except DCT=4x4, B=NO=8) , we require an output buffer to hold these data, and then we can write these data onto the output FIFO. This requires an FSM (this is actually an FSM that is a portion of the FSM @ clkfx in Fig. 4).

IV.2. Dynamic Region (PRR) I/O:

`dyn_inv(31..0) ← iFIFO_DO`	`oFIFO_DI ← dyn_outv(31..0)`
`dyn_inv(32) ← ofull`	`owren ← dyn_outv(32)`
`dyn_inv(33) ← iempty`	`irden ← dyn_outv(33)`
`dyn_inv(34) ← rst`	

IV.3. Finite State Machines:

IV.3.1. FSM @ PLB_Clk:

- FIFOs have to be reset prior to usage for at least 3 read/write clock cyles. If we use 16 cycles @ 100 MHz, then the minimum clkfx is 16x10ns/3 = 53.33ns →18.75MHz.

Figure 7 shows the FSM at PLB_clk. The register to 'fifo_fsm_rst' is just to avoid glitches (this is not a big deal, it was done to avoid simulation problem as a reset to a FIFO has to be clean).

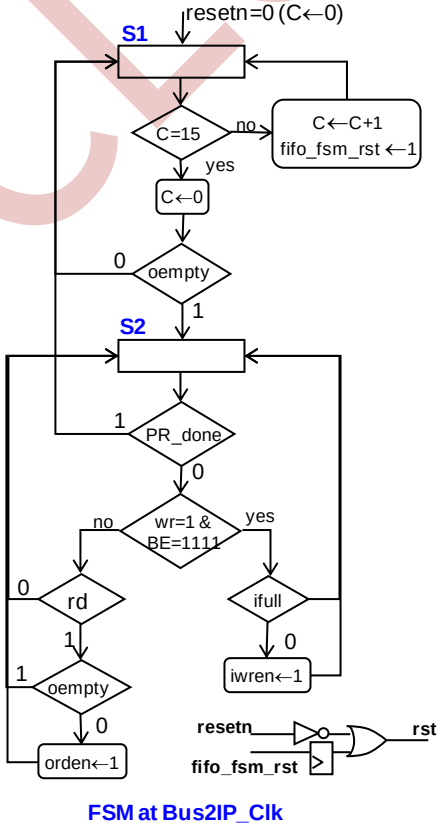

Figure 7. FSM at PLB_clk. 'C' represents a counter. 'resetn': low-level external reset

IV.3.2. FSM @ CLK_FX:

This is divided in two parts: (i) The FSM that controls the output buffer and 'owren', and (ii) the FSM that controls the input interface and output interface.

Figure 8 shows the FSM (at clk_fx) that ***controls the output buffer*** **and generates 'owren'**. When DCT=4x4 and B=NO=8, note that we do not require the output buffer and that owren=v.

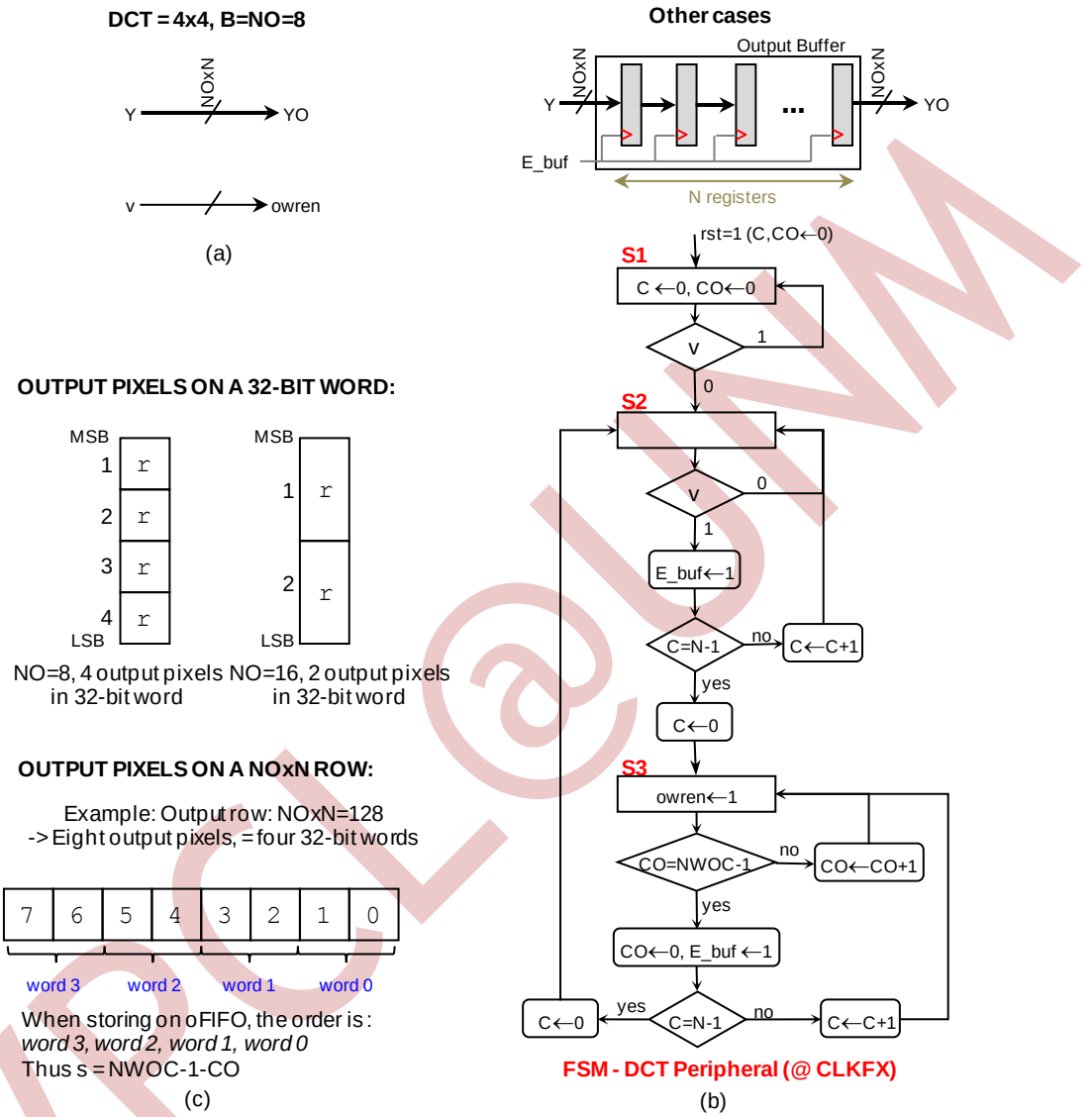

Figure 8. Output interface control and 'owren' generation. (a) owren=v and no output buffer when DCT=4x4, B=BO=8. (b) Output buffer and FSM that generates E_buf and owren. (c) Order of output pixels in a 32-bit word (same for input pixels) and on a NOxN output row.

Figure 9 depicts the timing diagram for the case DCT=8x8 and B=8, NO=16 (NWOC=4, NWIC=1)

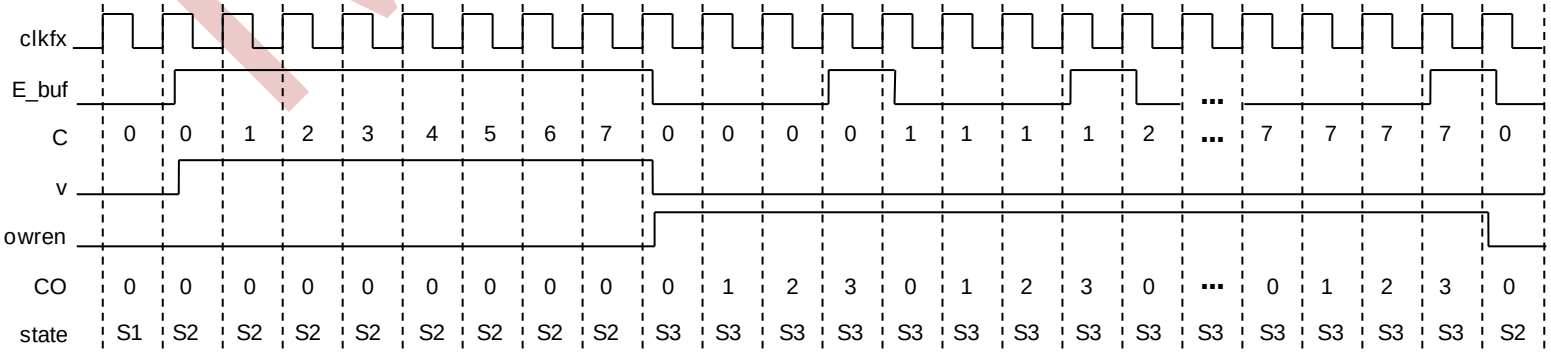

Figure 9. Timing diagram of the FSM of Fig. 7(b). DCT = 8x8, B=8, NO=16. Note that there are two stages. The first is the 2D DCT IP generating output data (generating eight v's). The second stage corresponds to storing the output buffer data on the oFIFO.

Now we deal with the FSM that ***controls the input and output interfaces***. The first signal to generate is Eri. The table below shows how Eri is generated for DCT=8x8 and 16x16. For 4x4, recall that Eri is not required:

DCT = 8x8			DCT = 16x16		
C	irden	Eri	C	irden	Eri
0	1	1	0	1	100
1	1	0	1	1	010
			2	1	001
			3	1	000

Based on the table, the following formula works for all N power of 2: $Eri = \frac{2^{(NWIC-1-C)}}{2(drop\ LSB)} AND\ irden$

TR=2: Fully pipelined case: Figure 10 depicts the FSM for every possible case of the DCT.

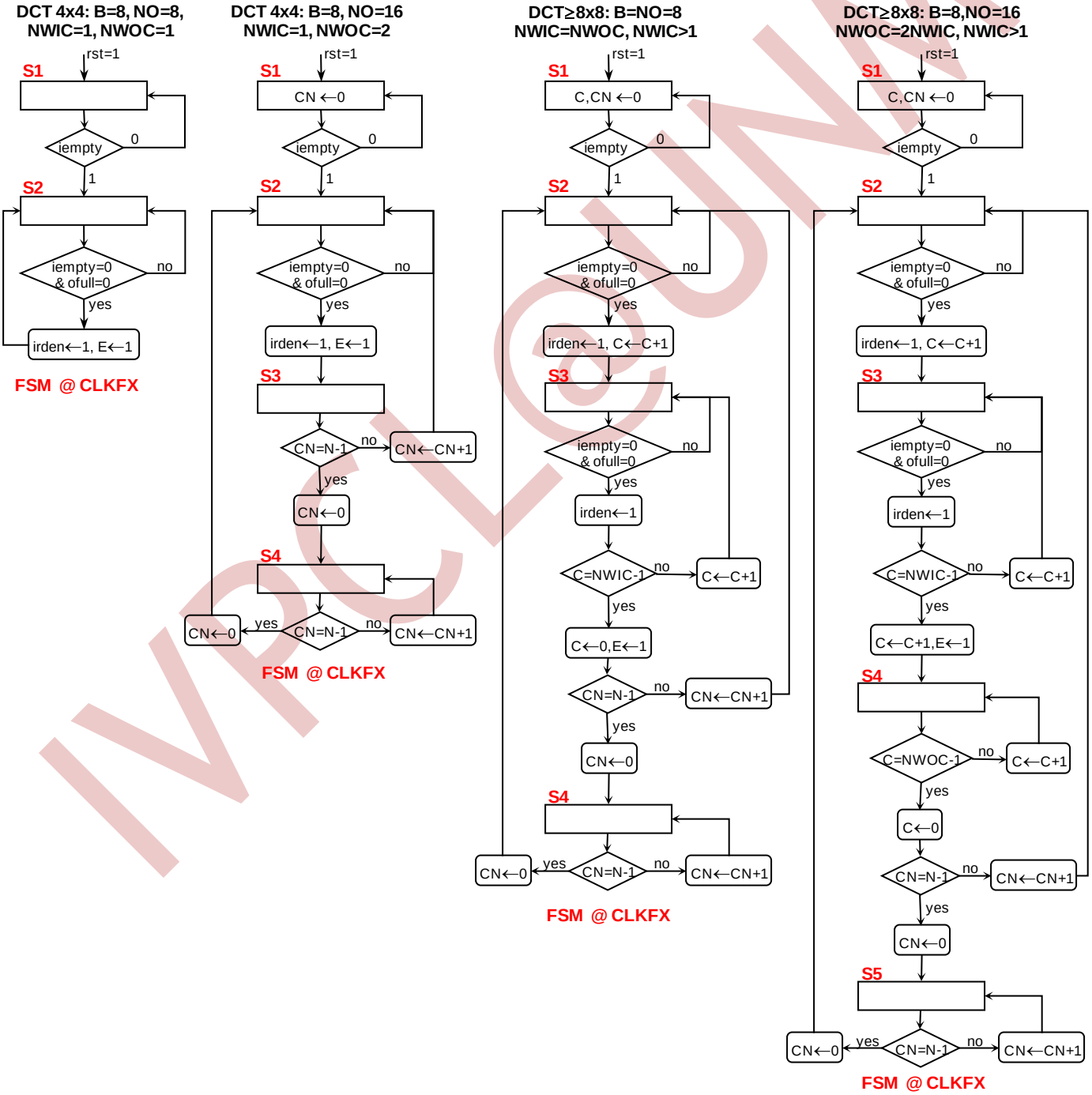

Figure 10. FSMs for the Fully Pipelined case (TR=2).

TR=1: One transpose case: Figure 11 depicts the FSM for every possible case of the DCT. Here we account for N-1 cycles in transpose buffer.

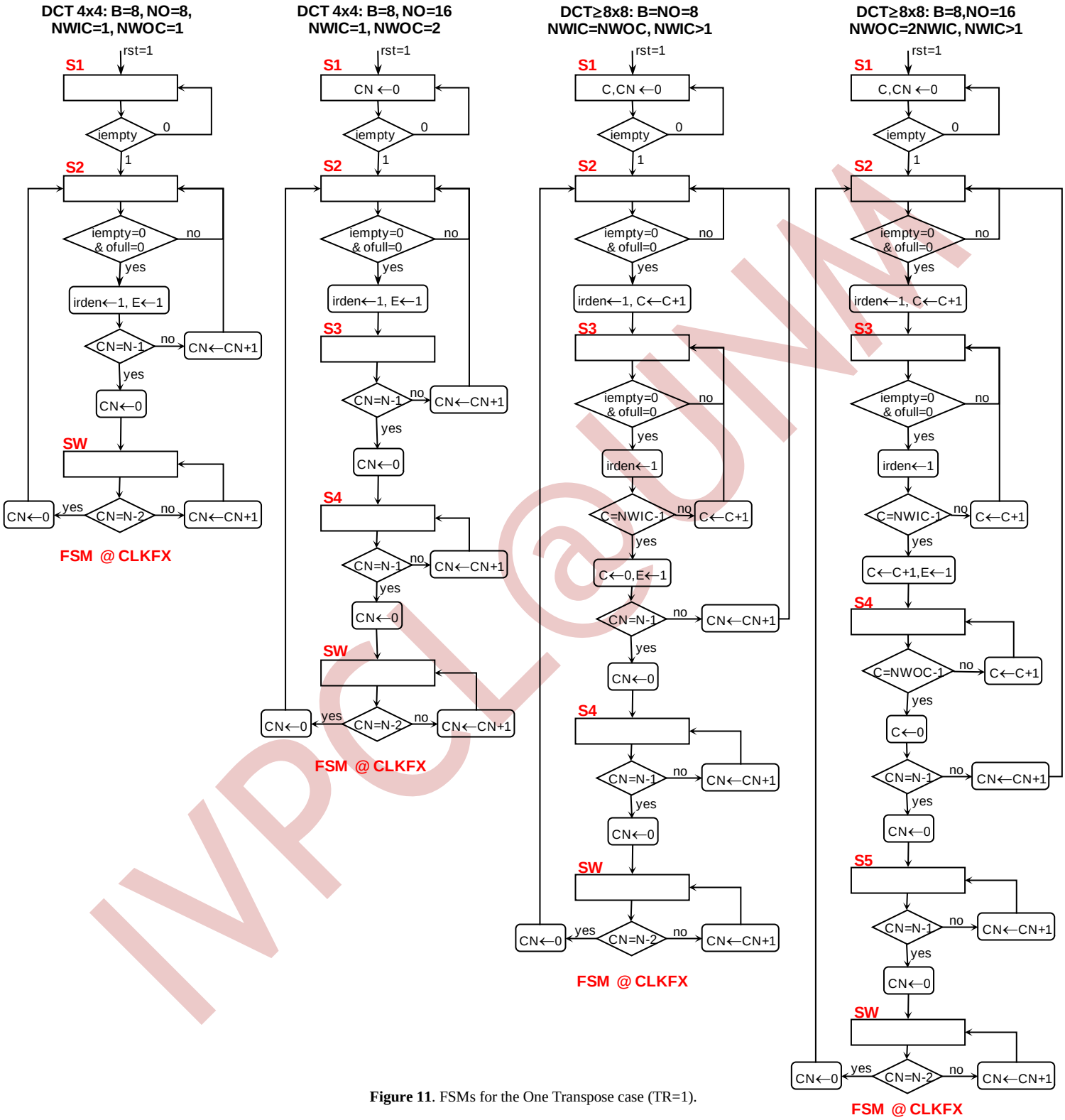

Figure 11. FSMs for the One Transpose case (TR=1).

IMPORTANT NOTES ABOUT VHDL PROJECT plb_ip_DCT2d_v2:
The COEFFICIENTS parameter has been already taken care of. The project comes with the text files containing the coefficients for the following cases:

- Even-odd decomposition: N=8,16, NH=10,12,14,16
- No decomposition: N=4, NH=10,12,14, 16