



IP TO IP INTERFACE

DESCRIPTION

The I_P2IP.VHD ipcore described here is designed to control iterative or combinational intellectual property VLSI components and interface them to other (non CPU) intellectual property components. It contains all the logic necessary to sequence the internal logic of a VLSI design and can be instantiated either inside or outside it. When instantiated outside, a single IP2IP component can control and interface multiple VLSI designs and in so doing addresses the inability of synthesis tools to remove duplicate registers completely.

Interface multiple simple multipliers to form a **Complex multiplier** or a **Power Calculator** .

Control and interface Simple **multipliers: Sequential, Array** and **Block memory**.

Provides interface and control logic for complex designs such as a **Cordic processor**.

APPLICATION - DSP, Power

VHDL Component Declaration:

```
COMPONENT I_P2IP
  GENERIC (
    WDN    : INTEGER :=13;
    WDL    : INTEGER :=9;
    ITR1   : INTEGER :=13;
    ITR2   : INTEGER :=1;
    ITRW   : INTEGER :=1;
    CLTI   : INTEGER :=0;
    PLSI   : INTEGER :=1;
    REPC   : INTEGER :=1;
    CLTA   : INTEGER :=1;
    CLTM   : INTEGER :=1;
    CLTS   : INTEGER :=1;
    RCW    : INTEGER :=2;
    DIFP   : INTEGER :=0;
    WDD    : INTEGER :=0;
    CLTW   : INTEGER :=0
  );
  PORT (
    BEG    : IN      NODE:='0';
    CLKI   : IN      NODE:='0';
    RST    : IN      NODE:='0';
    IOM    : BUFFER  BUS1D(16 DOWNT0 0);
    ECN    : BUFFER  BUS1D(WDN-1 DOWNT0 0);
    ECL    : BUFFER  BUS1D(WDL-1 DOWNT0 0);
    DNC    : BUFFER  BUS1D(RCW-1 DOWNT0 0);
    DNL    : BUFFER  BUS1D(WDD-1 DOWNT0 0);
    WRS    : BUFFER  BUS1D(ITRW-1 DOWNT0 0);
    ENI    : BUFFER  NODE
  );
END COMPONENT;
```

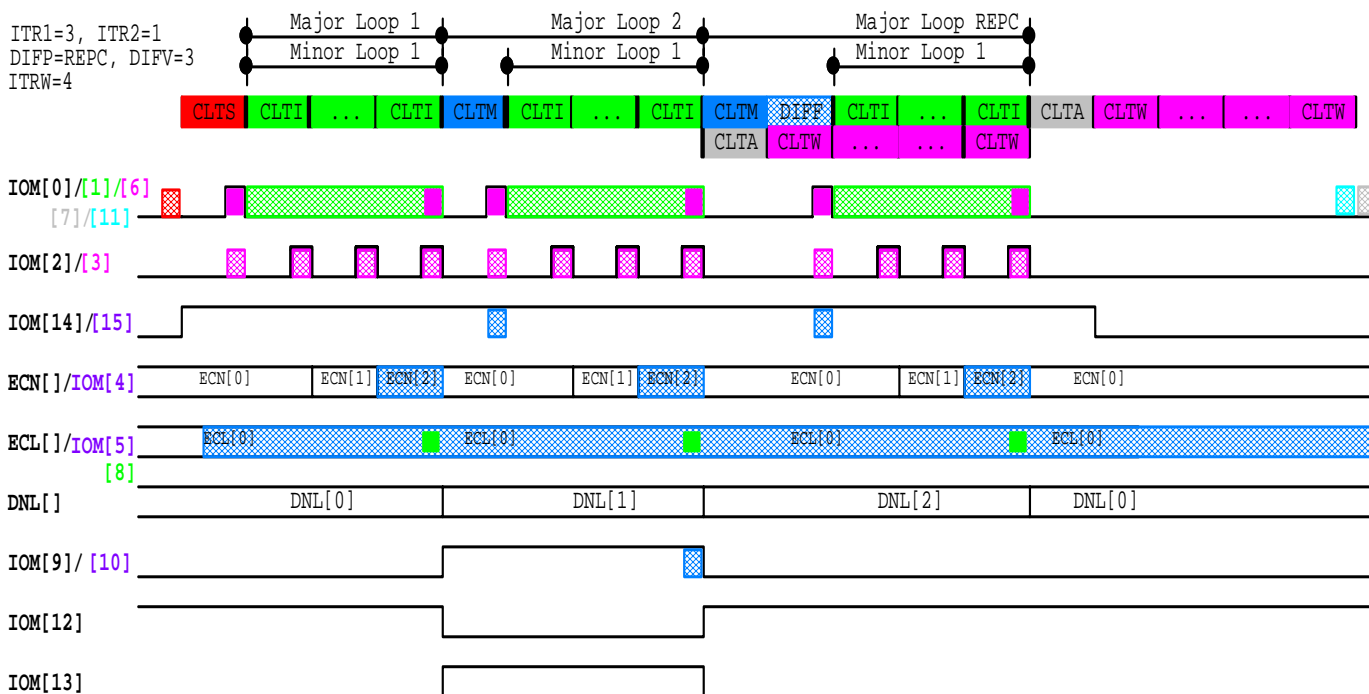
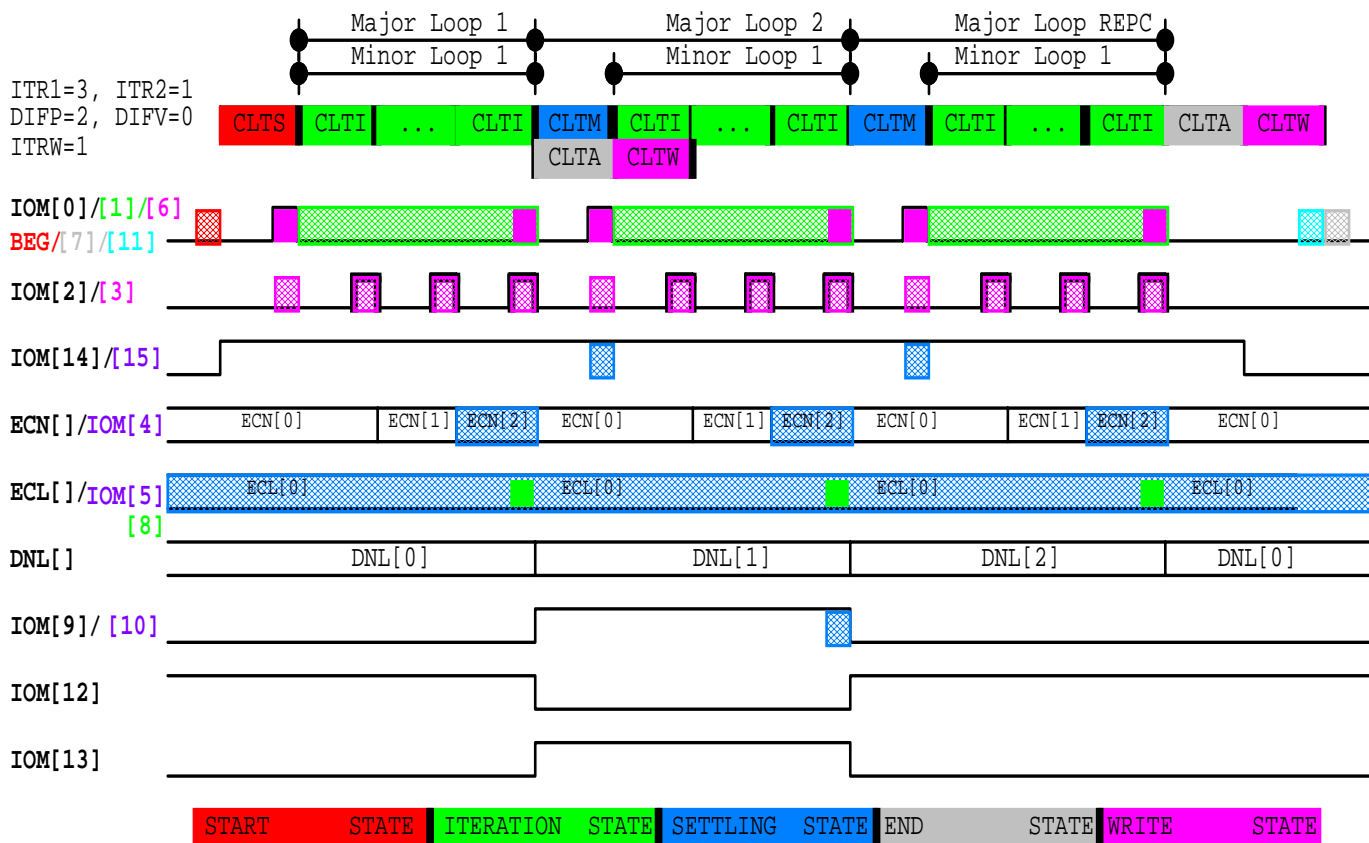


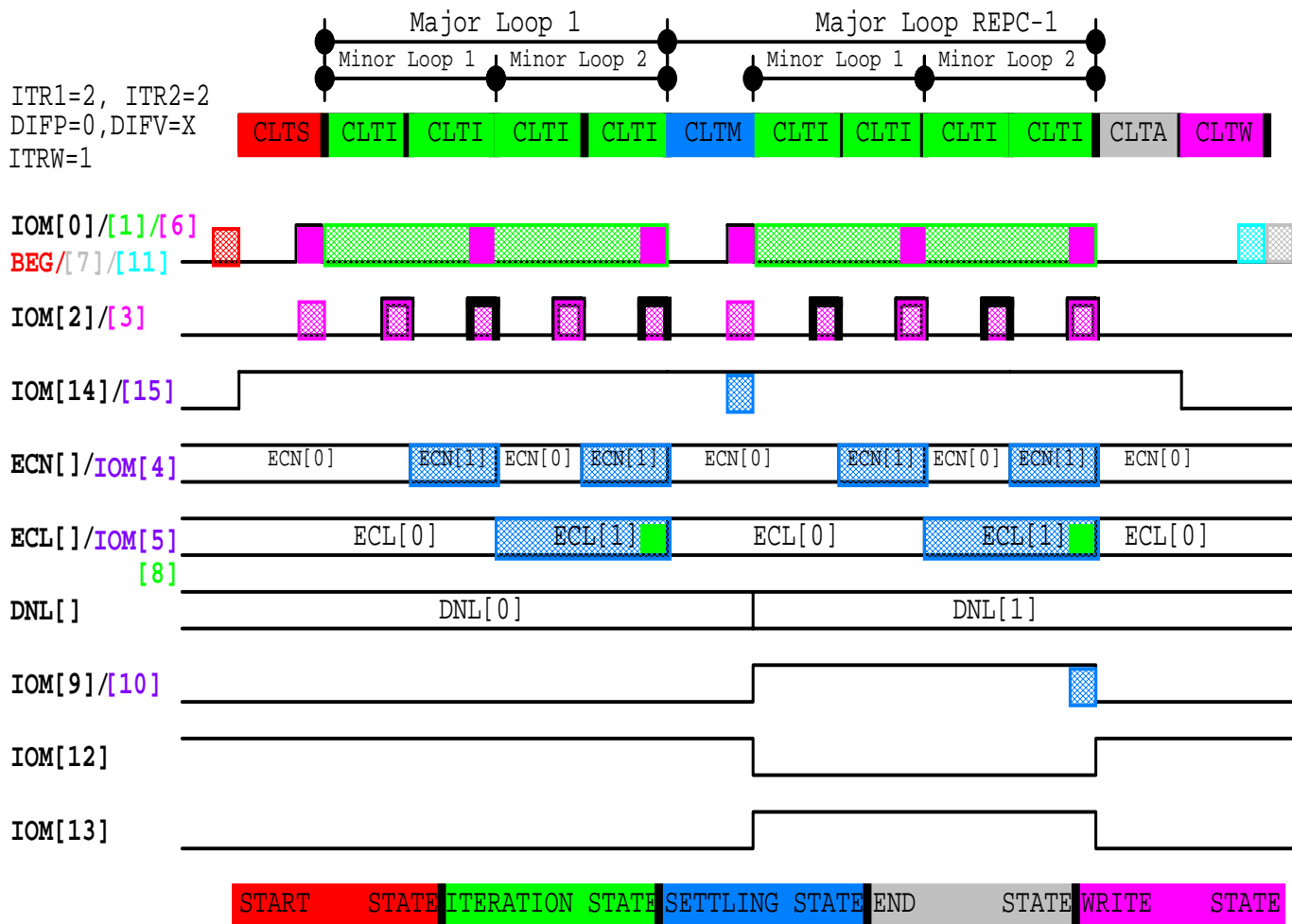
FILES YOU GET

i)FUNC.DOC	-	Documentation of functions & data types used in the core.
ii)README.DOC	-	Compile and licensing information.
iii)IP2IP.DOC	-	This document
MYLIB.VHD	-	PACKAGE
I_P2IP.VHD	-	TOP HIERARCHY DESIGN FILE
P_LSE.VHD	-	DESIGN FILE BELOW TOP HIERARCHY
S_DFF.VHD	-	-do-
F_DIV.VHD	-	-do-
U_DCNT.VHD	-	-do-
M_DFF.VHD	-	-do-
I_NCDEC.VHD	-	-do-
D_ECODE.VHD	-	-do-
A_DSB.VHD	-	-do-
S_JKF.VHD	-	-do-
S_TFF.VHD	-	-do-
P_AD.VHD	-	-do-
S_TATE.VHD	-	-do-

OPERATION

The process starts at the end of the 1st CLKI pulse of the BEG input and consists of REPC major loops. A major loop consists of ITR2 minor loops, each of ITR1 iterations and each making one run of a sequential process. A major loop starts with a START state of CLTS clks(1st loop) or MUX SETTling state of CLTM clks (2nd loop onwards) allowing the operands of the sequential process to settle. This is followed by the iteration states of CLTI clks. A loop counter increments at the end of the last iteration state and indicates the end of a major loop and the beginning of a new one. The last major loop is followed by an END state of CLTA clks and then a WRITE CYCLE of ITRW WRITE states each CLTW clks wide, followed by an END of PROCESS pulse. An END state and WRITE cycle combination can also be made to overlap a specifically marked major loop. A state is inserted only if its duration is more than 0.







INPUT PORTS

NAME	DESC	WIDTH	COMMENTS
BEG	Start	1	Active hi, load operands at the end of START state and begin main loop. See PLSI
CLKI	clock	1	Synchronizes all internal operations
RST	reset	1	Resets all internal registers

OUTPUT PORTS (except when specified, all are 1 bit wide)

NAME	DESC	COMMENTS
IOM(0)	Process Start	Pulse, before start of major loop (last clk, MUX SETTLING, START states)
IOM(11)	Process End	Clk pulse at the last CLKI of the END state.
IOM(7)	Process End+1	Clk pulse after the last clk of the END state.
IOM(14)	Process On	Hi from beginning of START state to the end of IOM(11).
ECN	Iteration Count	Decoded Iteration count in the Minor loop .0->ITR1-1. (WDL bits)
IOM(1)	Iteration On	ITERATION states of a Major loop are on
IOM(2)	Iteration End	Clk pulse at the end of each ITERATION state.
IOM(3)	Iteration Start/End	Clk pulse before start or at end, of ITERATION state. IOM(0) # (2)
IOM(4)	Iteration Last	Hi during the last ITERATION state of a minor loop
ECL	Minor loop Count	Decoded Minor loop count. 0->ITR2-1. Increment on ECN rollover (WDL bits)
IOM(5)	Minor loop Last	Hi during the ITERATION states of the last minor loop of a major loop.
IOM(6)	Minor loop Start/End	Clk pulse before start or at end, of Minor loop.
DNL	Major loop count	A counter counts from 0 to the Major Loop End Count (REPC-1), incrementing at the end IOM(8). DNL[] is the decoded count. WDD bits wide. Normally, WDD>=REPC. It is truncated or padded as necessary.
DNC	Major Loop count	O/P of loop counter mentioned in DNL. (RCW bits)
IOM(12)	Major loop Even	(REPC-1)-Major Loop count, is even.
IOM(13)	Major loop Middle	Current major loop is not 1st (DNL[0]) and not last (DNL[REPC-1])
IOM(8)	Major loop End	Clk pulse at last clk of the last iteration of a major loop. Major loop counter increments at end of pulse
IOM(9)	Marked Major Loop	Major Loop count DIFP is not on
IOM(10)	Marked Major Loop End	Major Loop count DIFP last CLKI
IOM(15)	Settling State End	1 CLKI wide pulse at the end of a MUX SETTLING state.
WRS	Write Iteration Count	A counter counts from 0 to the End Count (ITXW ⁴ -1), incrementing every CLTW clks. WRS of width, ITRW bits, is the decoded count ANDed with an internal "State On" signal.
IOM(16)	Write End	Clk pulse at the end of each WRITE state.

PARAMETERS (All integer type, with min value=0)

NAME	DESCRIPTION
WDL	Width of ECN o/p. WDL>=ITR1-1. If greater, padded w/ '0's, else truncated.
WDL	Width of ECL o/p. WDL>=ITR2-1. If greater, padded w/ '0's, else truncated.
RCW	Width of DNC o/p. RCW>=1+Log2(REPC-1). If greater, padded w/ '0's, else truncated
WDD	Width of DNL o/p. WDD>=REPC. If greater, padded w/ '0's, else truncated.
ITR1	iteration states/minor loop. Forced to >=1 when CLTI>0, else 0. Total iteration states=NITR ² .
ITR2	minor loops/major loop. Forced to >=1 when CLTI>0, else 0. Total iteration states=NITR ² .
ITRW	iteration states/Write Cycle. Forced to >=1 when CLTW>0, else ITRW
CLTI	Clocks in one iteration of the Sequential Process. CLTI>=EXN ¹
PLSI	BEG width=1 CLKI pulse(1) or more (0). When '0', pulse conditioning logic is used.
REPC	Major loop End Count (number of Sequential Process repeats). REPC=0 is forced to 1
CLTA	Duration in CLKI pulses of the END state. When CLTA=0, this state is removed.
CLTM	Duration in CLKI pulses of the MUX SETTLING state. When CLTM=0, this state is removed.
CLTS	Duration in CLKI pulses of the START state. When CLTS=0, this state is removed.
CLTW	Duration in CLKI pulses of the WRITE state. When CLTW=0, this state is removed.
DIFP	A Major Loop marked for a parallel END and WRITE state combination to appear. The SETTLING state of a major loop of this number may be extended by DIFV ³ clks so that the end of the last ITERATION coincides with the end of the parallel WRITE, the reason being that the calculations performed in the ITERATION states may not remain stable after the last one. DIFP is ignored if EXN ¹ =0. If DIFP=3 (for example), DNL[2] will be hi when the marked Major Loop is in progress

Notes

- 0>REP =1, if REPC>1, else 0
- 1>EXN =1, if REP*DIFP>1, else 0
- 2>NITR=ITR1*ITR2
- 3>DIFV=EXN*(CLTA+CLTW-CLTM-(NITR*CLTI))
- 4>ITXW=0, if CLTW=0, else ITRW with ITXW(min)=1