PE Instruction Set

31:28	27	26:24	23:16	15:8	7:0
res	wb	opcode	waddr	raddr1	raddr0

res: reserved
wb: write back signal
opcode: complex operation code
waddr: write address of the data memory
raddr1: read address 1 of the data memory
raddr0: read address 0 of the data memory

Opcode:

000	001	010	011	101	110	111
LOAD	ADD	SUB	MUL	MAX	MULSUB	MULADD

All of the above are complex operations. (e.g. ADD: (a+jb) + (c+jd))

Examples:

Operations	Instructions in Hex
MUL R1, R0 (WB)	32’h07_80_01_00
MUL R3, R2 (WB)	32’h07_81_03_02
MUL R5, R4 (WB)	32’h07_82_05_04
ADD R129, R128	32’h01_83_81_80

Assuming the overlay is comprised of an array of 256 PEs and each PE has 4 DSP blocks. The instruction schedule can be found as follows (if running at 500MHz, 1 cycle = 2ns):

Cycle	Operation	Instruction
256*32	***Load input data[1]***	**Nil**
1*32	*Complex multiplication*	(a+jb)*(c+jd) → a’ + jb’; c’ + jd’
1*80	*FFT*	W_N(c’+jd’) → tmp_r + jtmp_i
1*80		a’ + jb’ + (tmp_r + jtmp_i) → a’’ + jb’’ a’ + jb’ - (tmp_r + jtmp_i) → c’’ + jd’’
1*16	*Square & compare*	a’’*a’’ + b’’*b’’ or c’’*c’’ + d’’*d’’
256	*Shift internal data[2]*	Nil
1*32	*Complex multiplication*	(a+jb)*(c+jd) → a’ + jb’; c’ + jd’
1*80	*FFT*	W_N(c’+jd’) → tmp_r + jtmp_i
1*80		a’ + jb’ + (tmp_r + jtmp_i) → a’’ + jb’’ a’ + jb’ - (tmp_r + jtmp_i) → c’’ + jd’’
1*16	*Square & compare*	a’’*a’’ + b’’*b’’ or c’’*c’’ + d’’*d’’
256	*Shift data*	Nil
…	…	…
256*32	***Fetch output data[1]***	**Nil**

Load input data, *Shift internal data* and *Fetch output data* do not require instructions. They are handled by the SIPO and PISO modules.

Total latency = 256*32 + (32 + 80 + 80 + 16 + 256)*256 + 256*32 = 135168 ns = 135 us