Instruction Set

20 Oct. 2020

32-bit instruction format

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 in RISC-V fashion:

Operations	Instructions in Hex
MUL R128, R1, R0	32'h07_80_01_00
MUL R129, R3, R2	32'h07_81_03_02
MUL R130, R5, R4	32'h07_82_05_04
ADD R129, R128 (output)	32'h01_00_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'
1*80 1*80	*FFT*	a' + jb' + W_N(c'+jd') → a'' + jb'' a' + jb' - W_N(c'+jd') → c'' + jd''
1*16	*Square*	a''*a'' + b''*b''
32	*Shift internal data[2]*	Nil
1*32	*Complex multiplication*	(a+jb)*(c+jd) → a' + jb'; c' + jd'
1*80 1*80	*FFT*	a' + jb' + W_N(c'+jd') → a'' + jb'' a' + jb' - W_N(c'+jd') → c'' + jd''
1*16	*Square*	a''*a'' + b''*b''
32	*Shift data*	Nil
…	…	…
256*32	***Fetch output data***	**Nil**

[1,2]*Load input data*, *Shift internal data* and *Fetch output data* do not require instructions. They are handled by the SIPO and PISO modules.

Latency = (256*32 + (32+2*80+16) + 255*(32+32+2*80+16) + 256*32) * 2ns = 0.156 ms

Q. How to do a MAX operation among all the outputs of the 256 PEs (after square operation)? Add logic fabrics after the 256 PEs?