Lab Assignment 10

Multiply instructions

- MUL, MLA
- SMULL, SMLAL, UMULL, UMLAL

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Cond				F		I	Opcode				S	Rn				Rd				Operand2													Instruction
1110				00		1	Opcode				S	Rn				Rd				RotSpec				Imm8								DP imm
						0														ShiftAmount: Imm5				shift		0	Rm					DP reg sh_I
																				Rs			0	type		1						DP reg sh_R

1110	00	0	000	A	S	Rd	Rn	Rs	1	00	1	Rm	MUL, MLA

Rd <= Rm * Rs

Rd <= Rm * Rs + Rn

1110	00	0	01	U	A	S	RdHi	RdLo	Rs	1	00	1	Rm	U|S MULL, MLAL

RdHi, RdLo <= Rm * Rs

RdHi, RdLo <= Rm * Rs + RdHi, RdLo

4 x 4 multiplication

$a_3\ a_2\ a_1\ a_0$ x b_0

$a_3\ a_2\ a_1\ a_0$ x b_1

$a_3\ a_2\ a_1\ a_0$ x b_2

$a_3\ a_2\ a_1\ a_0$ x b_3

4x4 unsigned

$0\ 0\ 0\ a_3\ a_2\ a_1\ a_0$ x b_0

$0\ 0\ a_3\ a_2\ a_1\ a_0$ x b_1

$0\ a_3\ a_2\ a_1\ a_0$ x b_2

$a_3\ a_2\ a_1\ a_0$ x b_3

4x4 signed

$a_3\ a_3\ a_3\ a_3\ a_2\ a_1\ a_0$ x b_0

$a_3\ a_3\ a_3\ a_2\ a_1\ a_0$ x b_1

$a_3\ a_3\ a_2\ a_1\ a_0$ x b_2

$a_3\ a_2\ a_1\ a_0$ x $-b_3$

4x4 unsigned

$0\ 0\ 0\ a_3\ a_2\ a_1\ a_0 \times b_0$

$0\ 0\ a_3\ a_2\ a_1\ a_0 \times b_1$

$0\ a_3\ a_2\ a_1\ a_0 \times b_2$

$a_3\ a_2\ a_1\ a_0 \times b_3$

4x4 signed

$a_3\ a_3\ a_3\ a_3\ a_2\ a_1\ a_0 \times b_0$

$a_3\ a_3\ a_3\ a_2\ a_1\ a_0 \times b_1$

$a_3\ a_3\ a_2\ a_1\ a_0 \times b_2$

$\overline{a_3}\ \overline{a_2}\ a_1\ a_0 \times b_3$

Common part

0	0	0	a_3	a_2	a_1	a_0	x	b_0
0	0	a_3	a_2	a_1	a_0		x	b_1
0	a_3	a_2	a_1	a_0			x	b_2
a_3	a_2	a_1	a_0				x	b_3

a_3	a_3	a_3	a_3	a_2	a_1	a_0	x	b_0
a_3	a_3	a_3	a_2	a_1	a_0		x	b_1
a_3	a_3	a_2	a_1	a_0			x	b_2
$\overline{a_3}$	$\overline{a_2}$	a_1	a_0				x	b_3

Signed, unsigned multiplication

Signed multiplication:

```
signal a_s, b_s : signed (31 downto 0);
signal p_s : signed (63 downto 0);
p_s <= a_s * b_s;
```

Unsigned multiplication:

```
signal a_u, b_u : unsigned (31 downto 0);
signal p_u : unsigned (63 downto 0);
p_u <= a_u * b_u;
```

Synthesizing multipliers

Signed multiplication:

```
signal a_s, b_s : signed (31 downto 0);
signal p_s : signed (63 downto 0);
p_s <= a_s * b_s;        -- uses 4 DSP48E1
```

Unsigned multiplication:

```
signal a_u, b_u : unsigned (31 downto 0);
signal p_u : unsigned (63 downto 0);
p_u <= a_u * b_u;        -- uses 4 DSP48E1
```

Combining results

```
a_s <= signed (operand1);
b_s <= signed (operand2);
a_u <= unsigned (operand1);
b_u <= unsigned (operand2);
p_s <= a_s * b_s;
p_u <= a_u * b_u;
result <= std_logic_vector(p_s) when instr = smull
          else std_logic_vector(p_u);
```

-- uses 7 DSP48E1, but fails during routing

Another way

```
x1 <= operand1(31) when instr = smull else ‘0’;
x2 <= operand2(31) when instr = smull else ‘0’;
operand1_33 <= x1 & operand1;
operand2_33 <= x2 & operand2;
a_s <= signed (operand1_33);
b_s <= signed (operand2_33);
p_s <= a_s * b_s;
result_66 <= std_logic_vector(p_s);
      -- uses 4 DSP48E1 !
```