Limen Alpha control signals

Arithmetic and logic unit

op_code				result <=
0	0	0	0	operand_l or operand_r;
0	0	0	1	operand_l or not operand_r;
0	0	1	0	operand_l and operand_r;
0	0	1	1	operand_l and not operand_r;
0	1	0	0	operand_l xor operand_r;
0	1	0	1	operand_l sll operand_r;
0	1	1	0	operand_l srl operand_r;
0	1	1	1	operand_l sra operand_r;
1	0	0	0	operand_l < operand_r;
1	0	0	1	operand_l << operand_r;
1	0	1	0	operand_l - operand_r;
1	0	1	1	operand_l + operand_r;
1	1	0	0	operand_l;
1	1	0	1	operand_r;
1	1	1	0	operand_l[15..8] & operand_r[7..0];
1	1	1	1	operand_r[15..8] & operand_l[7..0];

Condition tester

cond_type			jmp_ack <=
0	0	0	'0';
0	0	1	'1';
0	1	0	'1' when test_data != 0 else '0';
0	1	1	'1' when test_data == 0 else '0';
1	0	0	'1' when test_data << 0 else '0';
1	0	1	'1' when test_data <<= 0 else '0';
1	1	0	'1' when test_data >> 0 else '0';
1	1	1	'1' when test_data >>= 0 else '0';

Sign extender

inst_form			data_out <=
0	0	0	(11..0 => data_in[7]) & data_in[6..3];
0	0	1	(11..0 => data_in[7]) & data_in[6..3];
0	1	0	(11..0 => '0') & data_in[6..3];
0	1	1	(11..0 => data_in[7]) & data_in[6..3];
1	0	0	(7..0 => '0') & data_in[9..2] when data_in[0] == '0'
			else data_in[9..2] & (7..0 => '0');
1	0	1	(8..0 => data_in[9]) & data_in[9..3];
1	1	0	(5..0 => data_in[9]) & data_in[9..0];
1	1	1	(11..0 => data_in[7]) & data_in[6..3];