Test Cases				
Instruction	**Assembly Code**	**Expected Result**	**Test Condition**	**Success?**
MOV AL r0,#1 ADD AL r1,r0 SYS AL	4001 0110 9800	r1 = 1	Tests register dependence on Op2 register	yes
MOV AL r0,#4 ADD AL r0,r1 SYS AL	4004 0101 9800	r0 = 4	Tests register dependence on Rd register	yes
MUL S r0,#0 MOV EQ r1,#1 ADD AL r1,r0 SYS AL	4a00 4411 0110 9800	r1 = 1	Tests condition code dependence & register dependence with a buffer instruction	yes
MOV S r0,#0 MOV NE r1,#2 MOV AL r1,#4 MOV AL r0,r1 SYS AL	4200 4612 4014 4101 9800	r0 = 4	Tests condition code dependence (with CC not being met) & register dependence	yes
MOV AL r0,#2 MOV AL r1,#5 MOV S r2,#0 MUL EQ r0,r1 MOV AL r2,r0 SYS AL	4002 4015 4220 4d01 4120 9800	r2 = 10	Tests condition code (with CC being met) and tests the MUL operation	yes
MOV AL r0,#5 MOV AL r1,#1 STR AL r0,[r1] LDR AL r2,[r1] SYS AL	4005 4011 7901 3921 9800	r2 = 5	Tests memory dependence	yes
MOV AL r0,#6 MOV AL r1,#1 MOV S r2,#0 STR EQ r0,[r1] LDR AL r3,[r1] SYS AL	4006 4011 4220 7d01 3931 9800	r3 = 6	Tests memory dependence in tandem with conditional execution	yes

ADD AL r0, #6	0006			
ADD AL r1, #1	0011			
MOV AL r15, #6	40f6			
SYS AL	9800			
SYS AL	9800			
SYS AL	9800			
ADD AL r1, r0	0110			
ADD AL r0, #5	0005			
ADD AL r0, #6	0006			
ADD AL r0, #5	0005			
ADD AL r1, r0	0110	r0=22; r1=29	Jumps	yes
PRE AL #1	c001			
PRE AL #2	c002			
ADD AL r0, #1	0001		Two PRE's in a	
SYS AL	9800	r0=33	row	yes
TOF AL r0,#6	3006			
ITOF AL r1,#0	3010	reg0 = 4’x40c0		
ITOF AL r2,#-3	302d	reg1 = 4’x0		
ITOF AL r3,#-7	3039	reg2 = 4’xc040		
SYS AL	9800	reg3 = 4’c0e0	Basic ITOF	yes
	c002			
	300a			
ITOF AL r0,#42	cff8			
ITOF AL r1,#-117	301b	reg0 = 0x4228		
SYS AL	9800	reg1 = 0xc2ea	Large # ITOF	yes
ITOF AL r0,#6	3006			
MOV AL r1, #0	4010			
FTOI AL r2, r0	2920	reg0 = 4’x40c0		
SYS AL	9800	reg2 = 4’x0006	Basic FTOI	yes
	cff8			
ITOF AL r0,#-117	300b			
MOV AL r1, #0	4010			
FTOI AL r2, r0	2920	reg0 = 4’xc2ea	Negative and	
SYS AL	9800	reg2 = 4’xff8b	Large FTOI	yes
ITOF AL r0,#5	3005			
ITOF AL r1, #6	3016			
MULF AL r0, r1	5101	reg0 = 16’b0100000111110000		
FTOI AL r2, r0	2920	reg1 = 16’b0100000011000000		
SYS AL	9800	reg2 = 16’b0000000000011110	Basic MULF	yes

ITOF AL r0,#-10
ITOF AL r1,#7
MULF AL r0, r1
FTOI AL r2, r0
SYS AL
	cfff
	3006
	3017
	5101
	2920
	9800
		reg0 = 16’b1100001010001100
		reg1 = 16’b0100000011100000
		reg2 = 16’b1111111110111010
			MULF on a Positive and Negative Float	yes

ITOF AL r0,#-21
ITOF AL r1,#-20
MULF AL r0, r1
FTOI AL r2, r0
SYS AL
	cffe
	300b
	cffe
	301c
	5101
	2920
	9800
		reg0 = 16’b0100001111010010
		reg1 = 16’b1100000110100000
		reg2 = 16’b0000000110100100
			MULF on Two Large Negative Values	yes

ITOF AL r0,#5
RECF AL r1,r0
SYS AL
	3005
	6910
	9800
		reg0 = 16’b0100000010100000
		reg1 = 16’b0011101101001100
			RECF	does not match up with precise reciprocal value

ITOF AL r1, #-50
FTOI AL r0, r1
SYS AL
	cffc
	301e
	2901
	9800
		reg0 = 16'b1111111111001110
		reg1 = 16'b1100001001001000
			FTOI on -50	no; reg0[14] turns out to be equal to 0 instead of 1