Interrupt Test

Interrupt 0	(C0 00)

Pass: Register output, all 0

Halt Test

Halt	(00 00)
Interrupt 0	(C0 00)

Pass: no output

Move Test 1

Move R0 123	(B0 7B)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R0 == 123

Move Test 2 (register test)

Move R0 0	(B0 00)
Move R1 1	(B1 01)
Move R2 2	(B2 02)
Move R3 3	(B3 03)
Move R4 4	(B4 04)
Move R5 5	(B5 05)
Move R6 6	(B6 06)
Move R7 7	(B7 07)
Move R8 8	(B8 08)
Move R9 9	(B9 09)
Move R10 10	(BA 0A)
Move R11 11	(BB 0B)
Move R12 12	(BC 0C)
Move R13 13	(BD 0D)
Move R14 14	(BE 0E)
Move R15 15	(BF 0F)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output, every register holding it’s own index

Add Test: 0 + 1 = 1

Move R0 0	(B0 00)
Move R1 1	(B1 01)
Add R0 R1 R2	(10 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 1

Add Test: 0 + -1 = -1

Move R0 0	(B0 00)
Move R1 -1	(B1 FF)
Add R0 R1 R2	(10 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Add Test: 0 + 0 = 0

Move R0 0	(B0 00)
Add R0 R0 R2	(10 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

Add Test: 1 + 0 = 1

Move R0 1	(B0 01)
Move R1 0	(B1 00)
Add R0 R1 R2	(10 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 1

Add Test: -1 + 0 = -1

Move R0 -1	(B0 FF)
Move R1 0	(B1 00)
Add R0 R1 R2	(10 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Subtract Test: 0 – 1 = -1

Move R0 0	(B0 00)
Move R1 1	(B1 01)
Subtract R0 R1 R2	(50 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Subtract Test: 0 – -1 = 1

Move R1 -1	(B1 FF)
Subtract R0 R1 R2	(50 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 1

Subtract Test: 0 – 0 = 0

Subtract R0 R0 R2	(50 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

Subtract Test: 1 – 0 = 1

Move R0 1	(B0 01)
Subtract R0 R1 R2	(50 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 1

Subtract Test: -1 – 0 = -1

Move R0 -1	(B0 FF)
Subtract R0 R1 R2	(50 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Multiply Test: 1 * 0 = 0

Move R0 1	(B0 01)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

Multiply Test: 0 * 1 = 0

Move R1 1	(B1 01)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

Multiply Test: 0 * 0 = 0

Multiply R0 R0 R2	(40 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

Multiply Test: 2 * 3 = 6

Move R0 2	(B0 02)
Move R1 3	(B1 03)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 6

Multiply Test: 2 * -1 = -2

Move R0 2	(B0 02)
Move R1 -1	(B1 FF)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2

Multiply Test: -1 * 2 = -2

Move R0 -1	(B0 FF)
Move R1 2	(B1 02)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2

Multiply Test: 2 * 1 = 2

Move R0 2	(B0 02)
Move R1 1	(B1 01)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 2

Multiply Test: 1 * 2 = 2

Move R0 1	(B0 01)
Move R1 2	(B1 02)
Multiply R0 R1 R2	(40 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 2

Divide Test: 2 / 1 = 2

Move R0 2	(B0 02)
Move R1 1	(B1 01)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 2

Divide Test: 2 / -1 = -2

Move R0 2	(B0 02)
Move R1 -1	(B1 FF)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2

Divide Test: -2 / -1 = 2

Move R0 -2	(B0 FE)
Move R1 -1	(B1 FF)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 2

Divide Test: 2 / 2 = 1

Move R0 2	(B0 02)
Divide R0 R0 R1	(30 01)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R1 == 1

Divide Test: -2 / -2 = 1

Move R0 -2	(B0 FE)
Divide R0 R0 R1	(30 01)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R1 == 1

Divide Test: 4 / 2 = 2

Move R0 4	(B0 04)
Move R1 2	(B1 02)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 1

Divide Test: -4 / 2 = -2

Move R0 -4	(B0 FC)
Move R1 2	(B1 02)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2

Divide Test: 5 / 2 = 2

Move R0 5	(B0 05)
Move R1 2	(B1 02)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 2

Divide Test: -5 / 2 = -2

Move R0 -5	(B0 FB)
Move R1 2	(B1 02)
Divide R0 R1 R2	(30 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2

Add Test: Max Positive Value

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == 2147483647 (Max positive signed 32-bit int)

Add Test: Min Negative Value

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Move R2 -1	(B2 FF)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Multiply R5 R2 R5	(45 25)	//R5 = 2147483647 * -1 = -2147483647
Add R6 R5 R6	(16 56)	//R6 = 0 + -2147483647 = -2147483647
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R6 == -2147483647 (Min negative signed 32-bit int)

Subtract Test: Max Positive Value

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Move R2 -1	(B2 FF)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Multiply R2 R5 R5	(42 55)	//R5 = -1 * 2147483647 = -2147483647
Subtract R6 R5 R6	(56 56)	//R6 = 0 - -2147483647 = 2147483647
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R6 == 2147483647

Subtract Test: Min Negative Value

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Subtract R6 R5 R6	(56 56)	//R6 = 0 - 2147483647 = -2147483647
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R6 == -2147483647

Multiply Test: Max Positive Value

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Multiply R1 R5 R5	(41 55)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == 2147483647

Multiply Test: Min Negative Value

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Move R2 -1	(B2 FF)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Multiply R2 R5 R5	(42 55)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == -2147483647

Divide Test: Max Positive Value / 1

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Divide R5 R1 R5	(35 15)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == 2147483647

Divide Test: Max Positive Value / Itself

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Divide R5 R5 R5	(35 55)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == 1

Divide Test: Min Negative Value / 1

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Move R2 -1	(B2 FF)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Multiply R2 R5 R5	(42 55)	//R5 = -1 * 2147483647 = -2147483647
Divide R5 R1 R5	(35 15)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == -2147483647

Divide Test: Min Negative Value / Itself

Move R0 64	(B0 40)	
Move R1 1	(B1 01)	
Move R2 -1	(B2 FF)	
Multiply R0 R0 R3	(40 03)	//R3 = 64 * 64 = 4096
Multiply R3 R0 R3	(43 03)	//R3 = 4096 * 64 = 262144
Multiply R3 R0 R3	(43 03)	//R3 = 262144 * 64 = 16777216
Multiply R3 R0 R3	(43 03)	//R3 = 16777216 * 64 = 1073741824
Subtract R3 R1 R4	(53 14)	//R4 = 1073741824 – 1 = 1073741823
Add R3 R4 R5	(13 45)	//R5 = 1073741824 + 1073741823 = 2147483647
Multiply R2 R5 R5	(42 55)	//R5 = -1 * 2147483647 = -2147483647
Divide R5 R5 R5	(35 55)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R5 == 1

And Test: All And None

Move R1 -1	(B1 FF)
And R1 R0 R2	(21 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

And Test: None And All

Move R1 -1	(B1 FF)
And R0 R1 R2	(20 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

And Test: None And None

And R0 R0 R2	(20 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

And Test: All And All

Move R1 -1	(B1 FF)
And R1 R1 R2	(21 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Or Test: All Or None

Move R1 -1	(B1 FF)
Or R1 R0 R2	(61 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Or Test: None Or All

Move R1 -1	(B1 FF)
Or R0 R1 R2	(60 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Or Test: None Or None

Move R1 -1	(B1 FF)
Or R0 R0 R2	(60 02)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == 0

Or Test: All Or All

Move R1 -1	(B1 FF)
Or R1 R1 R2	(61 12)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -1

Load Test

Load R2 R0 0	(82 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2113880064 (0x 8200 C000)

Load Test – Register Address

Move R1 8	(B1 08)
Load R2 R1 0	(82 10)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: R2 == -2112831488 (0x 8210 C000)

Load Test – Offset

Load R2 R0 1	(82 08)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass: R2 == -1073741824 (0x C000 0000)

Store Test

Move R0 0	(B0 00)	//These bits replaced with (00 00)
Move R1 -1	(B1 FF)	//These bits replaced with (00 FF)
Store R1 R0 0	(90 00)	//Overwrite previous 2 words
Interrupt 1	(C0 01)	
Halt	(00 00)	

Pass: Memory output reads: 0x **00 00 00 FF** 90 00 C0 01 00 00 …

Store Test – Register Address

Move R0 40	(B0 28)	
Move R1 -1	(B1 FF)	
Store R1 R0 0	(90 00)	
Interrupt 1	(C0 01)	
Halt	(00 00)	
		//These bits replaced with (00 00)
		//These bits replaced with (00 FF)

Pass: Memory output reads: 0x B0 50 B1 FF 90 00 C0 01 00 00 **00 00 00 FF** …

Store Test – Register Offset

Move R0 -1	(B0 FF)	
Store R0 R0 4	(90 04)	
Interrupt 1	(C0 01)	
Halt	(00 00)	
		//These bits replaced with (00 00)
		//These bits replaced with (00 FF)

Pass: Memory output reads: 0x B0 FF 90 04 C0 01 00 00 **00 00 00 FF** …

Pop Test

Move R0 -1	(B0 FF)	
Move R1 1	(B1 01)	
Pop R2	(A2 80)	
Interrupt 0	(C0 00)	
Halt	(00 00)	

Pass:
R2 == -1325420287 (0x B0FF B101)
R15 == 4

Push Test

Move R0 -1	(B0 FF)	//These bits replaced with (00 00)
Move R15 3	(BF 03)	//These bits replaced with (00 FF)
Push R0	(A0 40)	
Interrupt 1	(C0 01)	
Halt	(00 00)	

Pass: Memory output reads: 0x **00 00 00 FF** A0 40 C0 01 00 00 …

Return Test

Move R0 32	(B0 20)
Push R0	(A0 40)
Return	(A0 00)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R15 == 4

Fail: No output (failed to return past the first halt)

Jump Test

Move R0 -1	(B0 FF)
Jump 32	(77 00 00 20)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == -1

Fail: No register output (failed to jump past first halt)

Call Test

Call 24	(76 00 00 18)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R15 == 32

Fail: No register output (failed to call over first halt)

Branch Less Than – Less Branch Taken

Move R1 1	(B1 01)
BranchIfLess R0 R1 3	(70 01 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R1 == 1

Fail: No register output (failed to branch over first halt)

Branch Less Than – Less Branch Not Taken

BranchIfLess R0 R0 3	(70 00 00 03)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 0
Fail: No output (branched over Interrupt 0)

Branch Less or Equal – Less Branch Taken

Move R1 1	(B1 01)
BranchIfLessOrEqual R0 R1 3	(71 01 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R1 == 1
Fail: No output (failed to branch over first Halt)

Branch Less or Equal – Equal Branch Taken

BranchIfLessOrEqual R0 R0 3	(71 00 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R1 == 1
Fail: No output (failed to branch over first Halt)

Branch Less or Equal – Less or Equal Branch Not Taken

Move R0 1	(B0 01)
BranchIfLessOrEqual R0 R1 3	(71 01 00 03)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 1
Fail: No output (branched over Interrupt 0)

Branch if Equal – Equal Branch Taken

BranchIfLEqual R0 R0 3	(72 00 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 0
Fail: No output (failed to branch over first halt)

Branch if Equal – Equal Branch Not Taken

Move R1 1	(B1 01)
BranchIfLEqual R0 R1 3	(72 01 00 03)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R1 == 1
Fail: No output (branched over interrupt 0)

Branch if Not Equal – Not Equal Branch Taken

Move R1 1	(B0 01)
BranchIfNotEqual R0 R1 3	(73 01 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R1 == 1
Fail: No output (failed to branch over first halt)

Branch if Not Equal – Not Equal Branch Not Taken

BranchIfNotEqual R0 R0 3	(73 00 00 03)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 0
Fail: No output (branched over interrupt 0)

Branch if Greater – Greater Branch Taken

Move R0 1	(B0 01)
BranchIfGreater R0 R1 3	(74 01 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 1
Fail: No output (failed to branch over first halt)

Branch if Greater – Greater Branch Not Taken

BranchIfGreater R0 R0 3	(74 00 00 03)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 0
Fail: No register output (branched over interrupt 0)

Branch Greater or Equal – Greater Branch Taken

Move R0 1	(B0 01)
BranchIfGreater R0 R1 3	(75 01 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 1

Fail: No output (failed to branch over first halt)

Branch Greater or Equal – Equal Branch Taken

BranchIfGreater R0 R0 3	(75 00 00 03)
Halt	(00 00)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R0 == 0

Fail: No output (failed to branch over first halt)

Branch Greater or Equal – Greater or Equal Branch Not Taken

Move R1 1	(B1 01)
BranchIfGreater R0 R1 3	(75 01 00 03)
Interrupt 0	(C0 00)
Halt	(00 00)

Pass: Register output with R1 == 1

Fail: No output (branched over interrupt 0)

Checklist:

- Interrupt Test
- Halt Test
- Move
    - Move Test 1
    - Move Test 2 (Registers)
- Add
    - Add Test: 0 + 1 = 1
    - Add Test: 0 + -1 = -1
    - Add Test: 0 + 0 = 0
    - Add Test: 1 + 0 = 1
    - Add Test: -1 + 0 = -1
- Subtract
    - Subtract Test: 0 – 1 = -1
    - Subtract Test: 0 – -1 = 1
    - Subtract Test: 0 – 0 = 0
    - Subtract Test: 1 – 0 = 1
    - Subtract Test: -1 – 0 = -1
- Multiply
    - Multiply Test: 1 * 0 = 0
    - Multiply Test: 0 * 1 = 0
    - Multiply Test: 0 * 0 = 0
    - Multiply Test: 2 * 3 = 6
    - Multiply Test: 2 * -1 = -2
    - Multiply Test: -1 * 2 = -2
    - Multiply Test: 2 * 1 = 2
    - Multiply Test: 1 * 2 = 2
- Divide
    - Divide Test: 2 / 1 = 2
    - Divide Test: 2 / -1 = -2
    - Divide Test: -2 / -1 = 2
    - Divide Test: 2 / 2 = 1
    - Divide Test: -2 / -2 = 1
    - Divide Test: 4 / 2 = 2
    - Divide Test: -4 / 2 = -2
    - Divide Test: 5 / 2 = 2
    - Divide Test: -5 / 2 = -2
- Add Max
    - Add Test: Max Positive Value
    - Add Test: Min Negative Value
- Subtract Max
    - Subtract Test: Max Positive Value
    - Subtract Test: Min Negative Value
- Multiply Max
    - Multiply Test: Max Positive Value
    - Multiply Test: Min Negative Value
- Divide Max
    - Divide Test: Max Positive Value / 1
    - Divide Test: Max Positive Value / Itself

    - Divide Test: Min Negative Value / 1
    - Divide Test: Min Negative Value / Itself
- And
    - And Test: All And None
    - And Test: None And All
    - And Test: None And None
    - And Test: All And All
- Or
    - Or Test: All Or None
    - Or Test: None Or All
    - Or Test: None Or None
    - Or Test: All Or All
- Load
    - Load Test
    - Load Test – Register Address
    - Load Test – Offset
- Store
    - Store Test
    - Store Test – Register Address
    - Store Test – Register Offset
- Pop Test
- Push Test
- Return Test
- Jump Test
- Call Test
- Branch
    - Branch Less Than – Less Branch Taken
    - Branch Less Than – Less Branch Not Taken
    - Branch Less or Equal – Less Branch Taken
    - Branch Less or Equal – Equal Branch Taken
    - Branch Less or Equal – Less or Equal Branch Not Taken
    - Branch if Equal – Equal Branch Taken
    - Branch if Equal – Equal Branch Not Taken
    - Branch if Not Equal – Not Equal Branch Taken
    - Branch if Not Equal – Not Equal Branch Not Taken
    - Branch if Greater – Greater Branch Taken
    - Branch if Greater – Greater Branch Not Taken
    - Branch Greater or Equal – Greater Branch Taken
    - Branch Greater or Equal – Equal Branch Taken
    - Branch Greater or Equal – Greater or Equal Branch Not Taken