KEY, Edit Minamally

Jordan D. Ulmer && Patrick J. Schroeder

FA2014 (11/23/2014)

Processor Validation Spreadsheet

Kyle's Compiler Key

Format A ===
Format B ===
Format C ===

Format
Format
Format

OPName
OPName
OPName

Rdestination
Rdestination

Rsource1
Rsource1

Rsource2

ImmediateValue
ImmediateValue

Max IMMEDIATE

-32768<VAL<65536
-16777216<VAL<33554432

OP	RTL Instruction	Comments	Compiler Format	MIF Instruction	Instruction Format	OP	DEC Rdst	DEC Src1	DEC Src2	DEC Immediate	HEX Immediate	PC	Flags	HEX Contents Of The Register File R[0]	Conventionally RSRC1 R[1]	Conventionally RSRC2 R[2]	Conventionally RDST R[3]
NOP	-	Validate No Operation	A NOP R0,R0,R0	0000003F	A	NOP	0	0	0	-	-	0		0			
LDU#	R[1] <= 1	Validate LDU#	B LDU R1,R0,1	00400063	B	LDU	1	0	-	1	00000001	1		0	00000001		
LDU#	R[2] <= 2	-	B LDU R2,R0,2	008000A3	B	LDU	2	0	-	2	00000002	2		0	00000001	00000002	
ADD	R[3] <= R[1] + R[2]	Validate ADD Without An Overflow -- 16 Bit Immdiate Values Will Not Produce Overflow	A ADD R3,R1,R2	08860040	A	ADD	3	1	2	-	-	3		0	00000001	00000002	3
LD#	R[1] <= -3	Validate LD#	B LD R1,R0,-3	007FFF62	B	LD	1	0	-	-3	FFFFFFFD	4		0	FFFFFFFD		
LD#	R[2] <= -3	-	B LD R2,R0,-3	00BFFF62	B	LD	2	0	-	-3	FFFFFFFD	5		0	FFFFFFFD	FFFFFFFD	
SUB	R[3] <= R[1] - R[2]	Validate SUB (Zero Flag)	A SUB R3,R1,R2	08860100	A	SUB	3	1	2	-	-	6	ZERO_FLAG	0	FFFFFFFD	FFFFFFFD	0
LD#	R[1] <= -5	-	B LD R1,R0,-5	007FFEE2	B	LD	1	0	-	-5	FFFFFFFB	7		0	FFFFFFFB		
NEG	R[3] <= - R[1]	Validate Twos Complement Negation (Negative Flag)	A NEG R3,R1,R2	08860280	A	NEG	3	1	2	-	-	8	NEGATIVE_FLAG	0	FFFFFFFB	0	5
LDU#	R[1] <= 3	-	B LDU R1,R0,3	004000E3	B	LDU	1	0	-	3	00000003	9		0	00000003		
LDU#	R[2] <= 9	-	B LDU R2,R0,9	00800263	B	LDU	2	0	-	9	00000009	10		0	00000003	00000009	
AND	R[3] <= R[1] & R[2]	Validate Bitwise AnD	A AND R3,R1,R2	08860200	A	AND	3	1	2	-	-	11		0	00000003	00000009	1
LDU#	R[1] <= 9	-	B LDU R1,R0,9	00400263	B	LDU	1	0	-	9	00000009	12		0	00000009		
LDU#	R[2] <= 5	-	B LDU R2,R0,5	00800163	B	LDU	2	0	-	5	00000005	13		0	00000009	00000005	
OR	R[3] <= R[1] | R[2]	Validate Bitwise OR	A OR R3,R1,R2	08860240	A	OR	3	1	2	-	-	14		0	00000009	00000005	D
LDU#	R[1] <= 2	-	B LDU R1,R0,2	004000A3	B	LDU	1	0	-	2	00000002	15		0	00000002		
LDU#	R[2] <= 9	-	B LDU R2,R0,9	00800263	B	LDU	2	0	-	9	00000009	16		0	00000002	00000009	
XOR	R[3] <= XOR(R[1] , R[2])	Validate Bitwise XOR	A XOR R3,R1,R2	088602C0	A	XOR	3	1	2	-	-	17		0	00000002	00000009	B
LDU#	R[1] <= 2	-	B LDU R1,R0,2	004000A3	B	LDU	1	0	-	2	00000002	18		0	00000002		
LDU#	R[2] <= 0	-	B LDU R2,R0,0	00800023	B	LDU	2	0	-	0	00000000	19		0	00000002	00000000	
COMP	R[3] <= -R[1] -1	Validate Bitwise One's Complement -- Twos Comp Minus One	A COMP R3,R1,R0	08060300	A	COMP	3	1	0	-	-	20		0	00000002	00000000	FFFFFFFFD
LD#	R[1] <= -7	-	B LD R1,R0,-7	007FFE62	B	LD	1	0	-	-7	FFFFFFF9	21		0	FFFFFFF9	00000000	
LSL	R[3] <= R[1] <<	Validate Logical Shift Left One Bit	A LSL R3,R1,R0	080604C0	A	LSL	3	1	0	-	-	22		0	FFFFFFF9	00000000	FFFFFFFF2
LD#	R[1] <= -10	-	B LD R1,R0,-10	007FFDA2	B	LD	1	0	-	-10	FFFFFFF6	23		0	FFFFFFF6	00000000	
LSR	R[3] <= >> R[1]	Validate Logical Shift Right One Bit	A LSR R3,R1,R2	08860400	A	LSR	3	1	2	-	-	24		0	FFFFFFF6	00000000	FFFFFFFFB
LD#	R[1] <= -7	-	B LD R1,R0,-7	007FFE62	B	LD	1	0	-	-7	FFFFFFF9	25		0	FFFFFFF9	00000000	
ASL	R[3] <= R[1] <<<	Validate Arithmetical Shift Left One Bit	A ASL R3,R1,R0	080604C0	A	ASL	3	1	0	-	-	26		0	FFFFFFF9	00000000	FFFFFFFF2
LD#	R[1] <= -5	-	B LD R1,R0,-5	007FFEE2	B	LD	1	0	-	-5	FFFFFFFB	27		0	FFFFFFFB	00000000	
ASR	R[3] <= >>> R[1]	Validate Arithmetical Shift Right One Bit	A ASR R3,R1,R2	08860440	A	ASR	3	1	2	-	-	28		0	FFFFFFFB	00000000	#VALUE!
LDU#	R[1] <= 7	-	B LDU R1,R0,7	004001E3	B	LDU	1	0	-	7	00000007	29		0	00000007	00000000	
ROL	R[3] <= {R[1]{30:0},CARRY_FLAG} CARRY_FLAG <= R[1]{31}	Validate Rotate With Carry Left One Bit	A ROL R3,R1,R0	08060680	A	ROL	3	1	0	-	-	30		0	00000007	00000000	E | F
LDU#	R[1] <= 7	-	B LDU R1,R0,7	004001E3	B	LDU	1	0	-	7	00000007	31		0	00000007	00000000	
ROR	R[3] <= {CARRY_FLAG,R[1]{31:1}} CARRY_FLAG <= R[1]{30}	Validate Rotate With Carry Right One Bit	A ROR R3,R1,R0	08060640	A	ROR	3	1	0	-	-	32	CARRY_FLAG	0	00000007	00000000	3 | 80000003
LDU#	R[1] <= 1	-	B LDU R1,R0,1	00400063	B	LDU	1	0	-	1	00000001	33		0	00000001	00000000	
ADD#	R[3] <= R[1] + 4	Validate ADD Immediate Without An Overflow -- 16 Bit Immdiate Values Will Not Produce Overflow	B ADD R3,R1,4	08C00101	B	ADD	3	1	-	4		34		0	00000001	00000000	5
LD#	R[1] <= -3	-	B LD R1,R0,-3	007FFF62	B	LD	1	0	-	-3	FFFFFFFD	35		0	FFFFFFFD	00000000	

SUB#	R[3] <= R[1] - -3	Validate SUB (Zero Flag)	B SUB R3,R1,-3	08FFFF44	B	SUB	3	1	-	-3	FFFFFFFD	36	ZERO_FLAG	0	FFFFFFFD	00000000	0
LDU#	R[1] <= 15	-	B LDU R1,R0,15	004003E3	B	LDU	1	0	-	15	0000000F	37		0	0000000F	00000000	
AND#	R[3] <= R[1] & 45	Validate Bitwise AnD	B AND R3,R1,45	08C00B48	B	AND	3	1	-	45	0000002D	38		0	0000000F	00000000	D
LDU#	R[1] <= 9	-	B LDU R1,R0,9	00400263	B	LDU	1	0	-	9	00000009	39		0	00000009	00000000	
OR#	R[3] <= R[1] | 3	Validate Bitwise OR	B OR R3,R1,3	08C000C9	B	OR	3	1	-	3	00000003	40		0	00000009	00000000	B
LDU#	R[1] <= 2	-	B LDU R1,R0,2	004000A3	B	LDU	1	0	-	2	00000002	41		0	00000002	00000000	
XOR#	**R[3] <= XOR(R[1] , 12)**	**Validate Bitwise XOR**	**B XOR R3,R1,12**	**08C0030B**	**B**	**XOR**	**3**	**1**	**-**	**12**	**0000000C**	**42**		**0**	**00000002**	**00000000**	**E**
STA	MEM[120] <= R[3]	Validate Store Absolute	B STA R3,R0,120	00C01E10	B	STA	3	0	-	120	00000078	43		0	00000002	00000000	E
LDA	R[1] <= MEM[120]	Validate Load Absolute	B LDA R1,R0,120	00401E20	B	LDA	1	0	-	120	00000078	44		0	00000002	E	E
STIX	EA <= (R[1] + 118) MEM[3] <= R[EA]	Validate Store Indexed	B STIX R3,R1,118	08C01D91	B	STIX	3	1	-	118	00000076	45		0	00000002	E	E
LDU#	R[1] <= 1	-	B LDU R1,R0,1	00400063	B	LDU	1	0	-	1	00000001	46		0	00000001	E	E
LDIX	EA <= (R[1] + 119) R[1] <= MEM[EA]	Validate Load Indexed	B LDIX R1,R1,119	08401DE1	B	LDIX	1	1	-	119	00000077	47		0	E	E	E
LDU#	R[1] <= 50	-	B LDU R1,R0,50	00400CA3	B	LDU	1	0	-	50	00000032	48		0	00000032	E	E
MOVE	R[3] <= R[1]	Validate Move/Copy	A MOVE R3,R1,R2	08860800	A	MOVE	3	1	2	-	-	49		0	00000032	E	00000032
LDU#	R[1] <= 106	-	B LDU R1,R0,106	00401AA3	B	LDU	1	0	-	106	0000006A	50		0	0000006A	E	00000032
LBI	EA <= (R[1] + R[2])=120 R[3] <= MEM[EA]	Validate Load Base With Index	A LBI R3,R1,R2	08860840	A	LBI	3	1	2	-	-	51		0	0000006A	E	E
LDU#	R[2] <= 120	-	B LDU R2,R0,120	00801E23	B	LDU	2	0	-	120	00000078	52		0	0000006A	00000078	E
LDRi	EA <= (R[2]) R[1] <= MEM[EA]	Validate Load Register Indirect	A LDRi R1,R1,R2	08820880	A	LDRi	1	1	2	-	-	53		0	E	00000078	E
												54		0			
												55		0			
LDU#	R[1] <= 32	-	B LDU R1,R0,32	00400823	B	LDU	1	0	-	32	00000020	56		0	00000020	00000020	0
JMP	PC <= R[1]	Validate Load Indexed	A JMP R1,R1,R0	08021000	A	JMP	1	1	0	-	-	MEM[1]		0	00000020	00000020	0
												#VALUE!		0			
												#VALUE!		0			
LDU#	R[1] <= 32	-	B LDU R1,R0,32		B	LDU	1	0	-	32	00000020	#VALUE!		0	00000020	00000020	0
JMP	PC <= R[1]	Validate Jump PC to RSRC1	A JMP R1,R1,R0	00400823	A	JMP	1	1	0	-	-	MEM[1]		0	00000020	00000020	0
				08021000								#VALUE!		0			
												#VALUE!		0			
JSR	R[30] <= [PC] //R[30]=LINK PC <= MEM[1]	Validate Jump PC to Subroutine	A JSR R30,R1,R0	083C1040	A	JSR	30	1	0	-	-	MEM[1]		0	0	0	0
RTS	PC <= MEM[30] //R[30]=LINK	Validate Return From Subroutine	A RTS R1,R30,R0	F00210C0	A	RTS	1	30	0	-	-	MEM[30]		0	0	0	0
												#VALUE!		0			
												#VALUE!		0			
BEQ	if (R[1]=R[2]) EA<= 117 PC <= MEM[EA]	Validate Branch if Equal To	B BEQ R1,R2,117	10401D4C	B	BEQ	1	2	-	117	00000075	MEM[2]		0	0	0	0
BNE	if (NOT(R[1]=R[2])) EA<= 118 PC <= MEM[EA]	Validate Branch if NOT Equal To	B BNE R1,R2,118	10401D8A	B	BNE	1	2	-	118	00000076	MEM[2]		0	0	0	0
BLT	if (R[1]<R[2]) EA<= 119 PC <= MEM[EA]	Validate Branch if Less Than	B BLT R1,R2,119	10401DCF	B	BLT	1	2	-	119	00000077	MEM[2]		0	0	0	0
												#VALUE!		0			
												#VALUE!		0			
												#VALUE!		0			
BRA	EA<= 110 PC <= MEM[EA]	Validate Branch if Equal To	C BRA 110	00001BB0	C	BRA	-	-	-	110	0000006E	MEM[110]		0	0	0	0
BSR	if (R[-]<R[-]) EA<= 111 PC <= MEM[EA]	Validate Branch if Equal To	C BSR 111	00001BF1	C	BSR	-	-	-	111	0000006F	MEM[111]		0	0	0	0
												#VALUE!		0			
												#VALUE!		0			