Memory Map:

Address:	0x0000	0x0001	0x0002	0x0003	0x0004	0x0005	0x0006	0x0007	0x0008	0x0009	0x000A	0x000B	0x000C	0x000D	0x000E ...
RAM1:	Flag & Input	Program counter PC			Program counter reversed				Stack pointer SP		Stack value SPVAL				Registers and program data...
RAM2:	Flag	Program counter PC			Result's destination address		Instruction result				Stack and program data...				

Instructions

OP code	Name	Operands	Width	Flag	Cycles	Total	Description
0x00	INIT	-	-	clear	256	256	Wait for clock to stabilize, then initialize RAM ICs to sequential mode
0x01	RESET	-	-	clear	235	235	Set program counter PC = 0x040000 and stack pointer SP = 0x000A
0x02	-	-	-	-	158	414	Shadow instruction: Fetch
0x03	-	-	-	-	256	414	Shadow instruction: Fetch continuation
0x04	-	-	-	-	129	129	Shadow instruction: Increment program counter PC = PC + 3
0x05	-	-	-	-	129	129	Shadow instruction: Increment program counter PC = PC + 5
0x06	-	-	-	-	129	129	Shadow instruction: Increment program counter PC = PC + 7
0x07	-	-	-	-	129	129	Shadow instruction: Increment program counter PC = PC + 8
0x08	-	-	-	-	162	291	Shadow instruction: Copy 32 bit result
0x09	-	-	-	-	130	259	Shadow instruction: Copy 16 bit result
0x0A	-	-	-	-	113	113	Shadow instruction: Copy program counter
0x0B	-	-	-	-	167	296	Shadow instruction: Store to RAM indirect
0x0C	-	-	-	-	151	280	Shadow instruction: Store to RAM indirect
0x0D	-	-	-	-	173	587	Shadow instruction: Arithmetic instruction dispatch
0x0E	STF	-	-	set	132	546	Set FLAG
0x0F	CLF	-	-	clear	132	546	Clear FLAG
0x10	NOP	-	-	-	132	546	No operation
0x11	MOV	addr16 <- addr16	16	-	231	774	Move 16 bit value
0x12	MOVW	addr16 <- addr16	32	-	146	851	Move 32 bit value
0x13	INC	addr16 <- addr16	16	overflow	231	774	Increment
0x14	DEC	addr16 <- addr16	16	overflow	231	774	Decrement
0x15	COM	addr16 <- addr16	16	zero	231	774	1's complement (NOT)
0x16	NEG	addr16 <- addr16	16	zero	231	774	2's complement
0x17	LSL	addr16 <- addr16	16	overflow	233	776	Left shift (<<)
0x18	LSR	addr16 <- addr16	16	overflow	233	776	Right shift (>>)
0x19	ROL	addr16 <- addr16	16	overflow	233	776	Left shift with carry
0x1A	ROR	addr16 <- addr16	16	overflow	255	798	Right shift with carry
0x1B	ASR	addr16 <- addr16	16	overflow	235	778	Arithmetic right shift (keeps sign bit)
0x1C	REV	addr16 <- addr16	16	-	238	781	Bit reverse
0x1D	ADDI	addr16 <- addr16, val16	16	overflow	231	774	Add immediate
0x1E	ADCI	addr16 <- addr16, val16	16	overflow	231	774	Add immediate with carry
0x1F	SUBI	addr16 <- addr16, val16	16	overflow	231	774	Subtract immediate
0x20	SBCI	addr16 <- addr16, val16	16	overflow	231	774	Subtract immediate with carry
0x21	ANDI	addr16 <- addr16, val16	16	zero	231	774	Logical AND with immediate
0x22	ORI	addr16 <- addr16, val16	16	zero	231	774	Logical OR with immediate
0x23	XORI	addr16 <- addr16, val16	16	zero	231	774	Logical XOR with immediate
0x24	ADD	addr16 <- addr16, addr16	16	overflow	171	887	Add register
0x25	ADC	addr16 <- addr16, addr16	16	overflow	171	887	Add register with carry
0x26	SUB	addr16 <- addr16, addr16	16	overflow	171	887	Subtract register
0x27	SBC	addr16 <- addr16, addr16	16	overflow	171	887	Subtract register with carry
0x28	AND	addr16 <- addr16, addr16	16	zero	171	887	Logical AND with register
0x29	OR	addr16 <- addr16, addr16	16	zero	171	887	Logical OR with register
0x2A	XOR	addr16 <- addr16, addr16	16	zero	171	887	Logical XOR with register
0x2B	JMP	addr24	-	-	197	611	Jump to address
0x2C	CALL	addr24	32	-	221	748	Copy following instruction's address (PC + 4) and current FLAG to SPVAL, then jump
0x2D	RET	-	32	restore	138	552	Move SPVAL to PC & FLAG (effectively returns from CALL and restores previous FLAG)
0x2E	BRFS	addr24	-	-	160	625|574	Branch if FLAG set
0x2F	BRFC	addr24	-	-	160	625|574	Branch if FLAG cleared
0x30	BREQ	addr16, addr24	16	-	243	708|657	Branch if register is zero
0x31	BRNE	addr16, addr24	16	-	243	708|657	Branch if register is not zero
0x32	LDI	addr16 <- value16	16	-	81	624	Load 16 bit immediate
0x33	LDIW	addr16 <- value32	32	-	113	656	Load 32 bit immediate
0x34	LD	addr16 <- [addr16]	16	-	238	911	Indirect load 16 bits from address
0x35	LDB	addr16 <- [addr16]	8	-	238	911	Indirect load 8 bits from address, set upper 8 bits to 0
0x36	ST	[addr16] <- addr16	16	-	163	873	Indirect store 16 bits to address
0x37	STB	[addr16] <- addr16	8	-	163	857	Indirect store 8 bits to address
0x38	LD2W	[addr16]	32	-	256	799	Indirect load 32 bits from address in RAM2 to SPVAL register
0x39	LD2	[addr16]	16	-	224	767	Indirect load 16 bits from address in RAM2 to SPVAL register
0x3A	ST2W	[addr16]	32	-	256	799	Indirect store 32 bits from SPVAL register to address in RAM2
0x3B	ST2	[addr16]	16	-	224	767	Indirect store 16 bits from SPVAL register to address in RAM2
0x3C	LPM	addr16 <- [addr16]	16	-	211	884	Indirect load 16 bits from address in FLASH
0x3D	LPB	addr16 <- [addr16]	8	-	211	884	Indirect load 8 bits from address in FLASH, set upper 8 bits to 0
0x3E	OUT	addr16	8	-	252	795	Output 8 bits over SPI
0x3F	HALT	-	-	clear	14	428	Stop execution