Instruction set

Simple commands (no parameters)

- **HLT** – Halt Processor Clock
- **NOP** – No Operation (do nothing)
- **CLC** – clear carry flag
- **STC** – set carry flag
- **CII** – clear interrupt inhibit (enable interrupts)
- **SII** – set interrupt inhibit (disable interrupts)
- **RTS** – return from subroutine
- **RTI** – return from interrupt

Instruction set

Jump commands (destination address as parameter)

- **JMP** <dest> – unconditional jump
- **JSR** <dest> – jump to subroutine
- **JCS**/**JNC** <dest> – jump if carry flag set/not set
- **JZS**/**JNZ** <dest> – jump if zero flag set/not set
- **JNS**/**JNN** <dest> – jump if negative flag set/not set
- **JVS**/**JNV** <dest> – jump if overflow flag set/not set

Stack commands

- **PSF**/**PLF** – Push/Pull Flag state onto/from Stack
- **PSH**/**PUL** <reg> – Push/Pull register value onto/from Stack

Move command (two addresses as parameters)

- **MOV** <dest>, <src> – copy source value to destination address
- **MOV** SP, RCD – set StackAddress from Register C and Register D
- **MOV** RCD, SP – get StackAddress into Register C and Register D

Instruction set

Input/Output commands

- **OUT** <src> – Show byte on single byte LED Display
- **OUT** <port>, <src> – Write Byte to I/O port
- **INP** <dest>, <port> – Read Byte from I/O port

ALU commands

- **ADD** <dest>, <src> - dest = dest + src
- **SUB** <dest>, <src> - dest = dest - src
- **AND** <dest>, <src> - dest = dest & src
- **OR** <dest>, <src> - dest = dest v src
- **XOR** <dest>, <src> - dest = dest ^ src
- **LSR** <dest> – logical shift right
- **LSL** <dest> – logical shift left
- **CMP** <dest>, <src> - dest - src (flags only)

Addressing modes

Register
- RA, RB, RC, RD – register value

```
mov RA, RB
```

Immediate
- #<byte> - direct 8-bit value

```
mov RB, #2
```

Absolute
- <addr> - 16-bit memory address

```
mov RA, 0xfffc
```

Register indirect
- [RCD] – 16-bit address made of Register C and Register D

```
mov RA, [RCD]
```

Absolute indexed
- <addr>,RB – 16-bit memory address plus register B

```
mov RA, 0xff00,RB
```

Indirect indexed
- [<addr>],RB – pointer to 16-memory address plus register B

```
mov RA, [0xff00],RB
```

Instruction set

Instruction count

3 Single parameter ALU commands (LSL, LSR, OUT)

- 6 addressing modes: 4 registers, absolute, register indirect

=> 3*6 = 18 instructions

6 ALU operations with two parameters (ADD, SUB, AND, OR, XOR, CMP)

- 4 destination registers
- 7 addressing modes: 4 registers, immediate, absolute, register indirect

=> 6*4*7 = 168 instructions (without Rx, Rx => 144)

Instruction set

Instruction count

Load operation (move value into register)
- 4 destination registers
- 7 addressing modes: 4 registers, immediate, absolute, register indirect

=> 7*4 = 28 instructions (without Rx, Rx => 24)

Store operation (move register value into memory)
- 4 source registers
- 2 addressing modes: absolute, register indirect

=> 4*2 = 8 instructions

Special load and store addressing modes (absolute indexed, indirect indexed)
- A-Register only

=> 2*2 = 4 instructions

Instruction set

Instruction count

10 Jump commands (JMP, JSR, JCS, JNC, JZS, JNZ, JNS, JNN, JVS, JNV)

- 2 addressing modes: absolute, register indirect

=> 10*2 = 20 instructions

Instruction set

Instruction count

2 Register stack operations (PSH, PUL)
- 4 registers

=> 2*4 = 8 instructions

4 Other stack operations
- PSF, PLF, MOV SP,RCD, MOV RCD,SP

=> 4 instructions

2 Port I/O operations (INP, OUT)
- 2 addressing modes (immediate, register)

=> 2*2 = 4 instructions

8 Other operations
- NOP, HLT, STC, CLC, SII, CII, RTS, RTI

=> 8 instructions

Instruction set

Instruction count

Instruction Type	Number
single parm ALU	18
two parm ALU	144
load	24
store	8
special address	4
jump	20
register stack	8
other stack	4
port I/O	4
other operations	8
total instructions	**242**