Simple Microprocessor Design - Instruction Set:

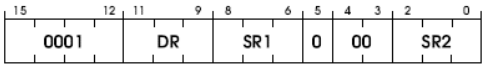
- **LC3-B ISA. 16-bit fixed ISA, 16 bits 8 registers,**
- **Byte access 2KB Memory**
- **4 instructions: ADD, BR, LDW , STW**

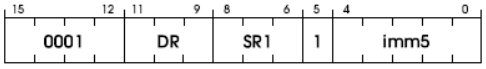
ADD

Assembler Formats

```
ADD    DR, SR1, SR2
ADD    DR, SR1, imm5
```

Encodings





Operation

```
if (bit[5] == 0)
  DR = SR1 + SR2;
else
  DR = SR1 + SEXT(imm5);
setcc();
```

Description

If bit [5] is 0, the second source operand is obtained from SR2. If bit [5] is 1, the second source operand is obtained by sign-extending the imm5 field to 16 bits. In both cases, the second source operand is added to the contents of SR1, and the result stored in DR. The condition codes are set, based on whether the result is negative, zero, or positive.

Examples

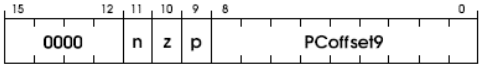
```
ADD    R2, R3, R4      ; R2 ← R3 + R4
ADD    R2, R3, #7      ; R2 ← R3 + 7
```

BR — Conditional Branch

Assembler Formats

BRn	LABEL	BRzp	LABEL
BRz	LABEL	BRnp	LABEL
BRp	LABEL	BRnz	LABEL
BR†	LABEL	BRnzp	LABEL

Encoding



Operation

```
if ((n AND N) OR (z AND Z) OR (p AND P))
   PC = PC‡ + LSHF(SEXT(PCoffset9), 1);
```

Description

The condition codes specified by the state of bits [11:9] are tested, as follows: If bit [11] is set, N is tested; if bit [11] is clear, N is not tested. If bit [10] is set, Z is tested, etc. If any of the condition codes tested is set, the program branches to the location specified by sign-extending the PCoffset9 field to 16 bits, left-shifting it one bit, and adding the result to the incremented PC. The PCoffset9 field specifies the number of instructions, forward or backwards, to branch over.

Examples

```
BRzp    LOOP    ; Branch to LOOP if the last result was zero or positive.
BR†     NEXT    ; Unconditionally Branch to NEXT.
```

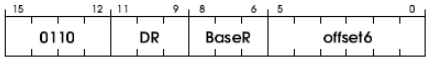
LDW

Load Word

Assembler Format

```
LDW     DR, BaseR, offset6
```

Encoding



Operation

```
DR = MEM[BaseR + LSHF(SEXT(offset6), 1)];
setcc();
```

Description

A word-aligned address is computed by sign-extending offset6 to 16 bits, left-shifting the result by one bit, and then adding this to the contents of the base register. The word starting at this address is stored into DR. The condition codes are set, based on whether the value loaded is negative, zero, or positive.

Example

```
LDW     R4, R2, #10      ; R4 ← MEM[R2 + 20]
```

Note

The base register must contain a word address (i.e., its contents must be even). If the base register contains an odd address, an illegal operand address exception occurs.

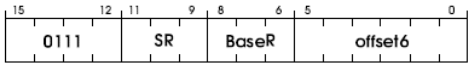
STW

Store Word

Assembler Format

```
STW    SR, BaseR, offset6
```

Encoding



Operation

```
MEM[BaseR + LSHF(SEXT(offset6), 1)] = SR;
```

Description

The contents of SR are stored into the word-aligned memory location whose address is obtained by sign-extending offset6 to 16 bits, left-shifting the result by one bit and adding this to the contents of the base register.

Example

```
STW    R4, R2, #10      ; MEM[R2 + 20] ← R4
```

Note

The base register must contain a word address (i.e., its contents must be even). If the base register contains an odd address, an illegal operand address exception occurs.