

microRISC

Andrew Schomber
December 11, 2024

Table of Contents

Architecture	<3>
→ Registers	<3>
→ → General Purpose Registers	<3>
→ → Program Counter (PC)	<3>
→ → Stack Pointer (SP)	<3>
→ → CMP Register	<3>
→ Opcode Layout and Control Signals	<4>
→ Memory Layout	<4>
Instruction Set	<5>
→ Arithmetic and Logical	<5>
→ Memory	<5>
→ Branching	<5>
→ Other	<6>

Architecture

Registers

General Purpose Registers

The general purpose registers are used to store data and perform arithmetic operations. They are named R0-R29 and are 32 bits wide. R30 is reserved for the stack pointer (SP) and R31 is reserved for the comparison register (CMP).

Register	Binary Representation	Register	Binary Representation
R0	00000	R1	00001
R2	00010	R3	00011
R4	00100	R5	00101
R6	00110	R7	00111
R8	01000	R9	01001
R10	01010	R11	01011
R12	01100	R13	01101
R14	01110	R15	01111
R16	10000	R17	10001
R18	10010	R19	10011
R20	10100	R21	10101
R22	10110	R23	10111
R24	11000	R25	11001
R26	11010	R27	11011
R28	11100	R29	11101
R30 (SP)	11110	R31 (CMP)	11111

Program Counter (PC)

The program counter register keeps track of the current instruction being executed. It is automatically incremented after each instruction is executed. It can not be directly accessed or modified by the programmer.

Stack Pointer (SP)

The stack pointer register is used to keep track of the top of the stack. Similarly to ARM, you must manually manage the stack pointer. Reference it using the SP keyword.

Register	Binary Representation
SP	11110

CMP Register

The CMP register is used to store the result of a comparison operation. It is set by the CMP instruction, which subtracts the second operand from the

first operand and sets the CMP register based on the result. Reference it using the CMP keyword (its an operation and a register).

Register	Binary Representation
CMP	11111

Opcode Layout and Control Signals

Every opcode is 6 bits wide. Each bit will be a dedicated control signal:

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5
Control Signal 0	Control Signal 1	Control Signal 2	Control Signal 3	Control Signal 4	Control Signal 5

There are also other control signals implied by the opcode. These are:

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5
Arithmetic and Logic	Memory	Branch	Other		

There are also other control signals that are not part of the opcode. These are:

Memory Layout

This is the content of Subsection 1.3.

Instruction Set

Arithmetic and Logical

The following arithmetic and logical operations are supported:

Syntax	Opcode	Rd	Rn	Rm
ADD Rd, Rn, Rm	000001	5 bits	5 bits	5 bits
SUB Rd, Rn, Rm	000011	5 bits	5 bits	5 bits
MUL Rd, Rn, Rm	000101	5 bits	5 bits	5 bits
DIV Rd, Rn, Rm	000111	5 bits	5 bits	5 bits
AND Rd, Rn, Rm	001001	5 bits	5 bits	5 bits
ORR Rd, Rn, Rm	001011	5 bits	5 bits	5 bits
XOR Rd, Rn, Rm	001101	5 bits	5 bits	5 bits
LSL Rd, Rn, Rm	001111	5 bits	5 bits	5 bits
LSR Rd, Rn, Rm	010001	5 bits	5 bits	5 bits
ASR Rd, Rn, Rm	010011	5 bits	5 bits	5 bits

Syntax	Opcode	Rd	Unused
NEG Rd	010101	5 bits	21 bits

Syntax	Opcode	Rd	Rn	Unused
MOV Rd, Rn	111111	5 bits	5 bits	16 bits

Syntax	Opcode	Rd	Imm
MOV Rd, Imm	111111	5 bits	21 bits

Memory

The following memory instructions are supported:

Syntax	Opcode	Rd	Rn	Rm
LDR Rd, [Rn, Rm]	000010	5 bits	5 bits	5 bits
STR Rd, [Rn, Rm]	000110	5 bits	5 bits	5 bits

Syntax	Opcode	Rd	Label	Unused
ADR Rd, Label	001010	5 bits	20 bits	1 bit

Branching

Labels are resolved to 24-bit addresses from the start of the program. The following branching instructions are supported:

Syntax	Opcode	Label	Unused
B Label	000100	24 bits	2 bits
BL Label	001100	24 bits	2 bits

Syntax	Opcode	Label	Unused
BEQ Label	010100	24 bits	2 bits
BNE Label	011100	24 bits	2 bits
BGT Label	100100	24 bits	2 bits
BLT Label	101100	24 bits	2 bits
BGE Label	110100	24 bits	2 bits
BLE Label	111100	24 bits	2 bits

Syntax	Opcode	Rd	Rn	Unused
CMP Rd, Rn	100000	5 bits	5 bits	16 bits

Syntax	Opcode	Rd	Label	Unused
CBZ Rd, Label	110000	5 bits	20 bits	1 bit
CBNZ Rd, Label	101000	5 bits	20 bits	1 bit

Syntax	Opcode	Unused
RET	111000	26 bits

Other

These are other instructions that don't fit under the existing categories:

Syntax	Opcode	Unused
NOP	000000	26 bits

There is also support for single line comments and end of line comments:

```
// This is a single line comment
ADD R1, R2, R3 // This is an end of line comment
```