

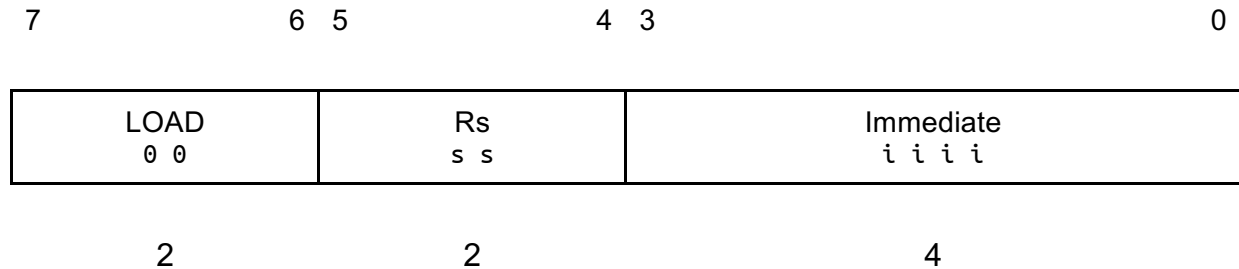
# CPEG324 Instruction Set

March 7th, 2016

Sean Krail

# LOAD

## Load Immediate Signed Byte



Format:       LOAD Rs, Immediate

Purpose:       To load a 4-bit constant into an 8-bit GPR.

Description:   Rs  $\leftarrow$  Immediate

              The 4-bit signed *Immediate* is loaded into GPR *Rs*.

Restrictions:

              The *Immediate* value must be an integer in the range of [-8, 8). Using an *Immediate* value outside of this range leads to undefined behavior.

Operation:

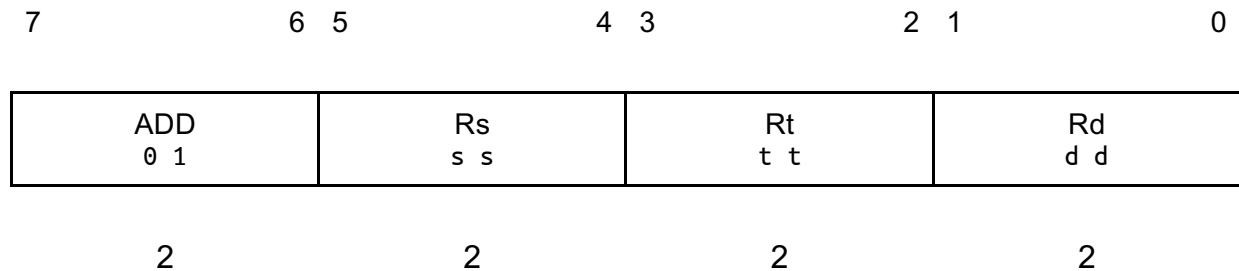
              GPR[Rs]  $\leftarrow$  Immediate

Exceptions:

              None

# ADD

Add Signed Byte



Format:        ADD Rd, Rs, Rt

Purpose:        To add 8-bit integers.

Description:    $Rd \leftarrow Rs + Rt$

The 8-bit byte value in GPR *Rt* is added to the 8-bit value in GPR *Rs* to produce an 8-bit result. If the addition results in 8-bit 2's complement arithmetic overflow then the destination register is not modified and an Integer Overflow exception occurs. If it does not overflow, the 8-bit result is placed into GPR *Rd*.

Restrictions:

None

Operation:

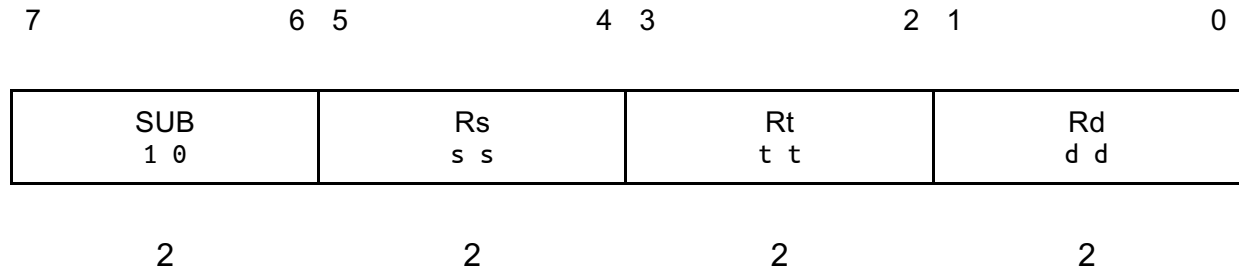
```
temp ← GPR[Rs] + GPR[Rt]
if (POSITIVE(GPR[Rs]) == POSITIVE(GPR[Rt]) && POSITIVE(temp) !=
POSITIVE(GPR[Rs])) then
    SignalException(IntegerOverflow)
else
    GPR[Rd] ← temp
endif
```

Exceptions:

Integer Overflow

# SUB

Subtract Signed Byte



Format: SUB Rd, Rs, Rt

Purpose: To subtract 8-bit integers.

Description:  $Rd \leftarrow Rs - Rt$

The 8-bit byte value in GPR *Rt* is subtracted from the 8-bit value in GPR *Rs* to produce an 8-bit result. If the subtraction results in 8-bit 2's complement arithmetic overflow, or if either of the source register's or target register's overflow flag is set, then an Integer Overflow exception occurs and the destination register's overflow flag is set. If it does not overflow, the 8-bit result is placed into GPR *Rd*.

Restrictions:

None

Operation:

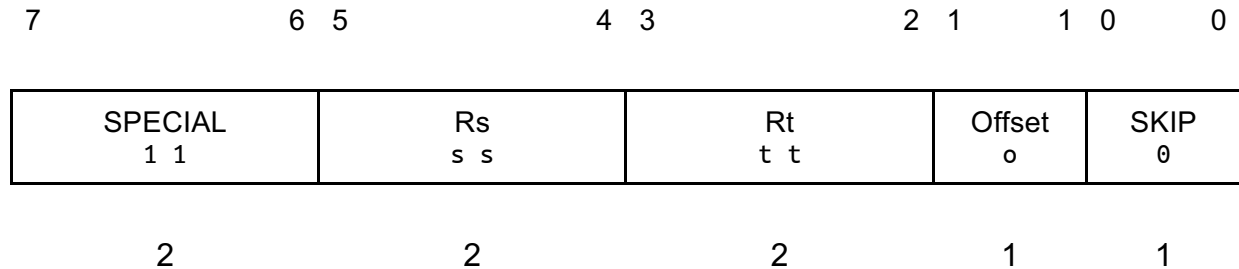
```
if (GPR[Rt] == -128) then
    SignalException(IntegerOverflow)
else
    temp1 ← (GPR[Rt] - 1) ^ 0xFF
    temp2 ← GPR[Rs] + temp1
    if ((POSITIVE(GPR[Rs]) == POSITIVE(temp1) && POSITIVE(temp2) !=
    POSITIVE(GPR[Rs])) || OVERFLOW[Rs] || OVERFLOW[Rt]) then
        SignalException(IntegerOverflow)
    endif
    GPR[Rd] ← temp2
endif
```

Exceptions:

Integer Overflow

# SKIP

Branch on Equal



Format: SKIP Rs, Rt, Offset

Purpose: To compare GPRs then do a PC-relative conditional branch.

Description: if (Rs = Rt) then branch

A 2-bit offset (the 1-bit *Offset* field plus 2) is added to the address of the instruction following the branch (not the branch itself), in the branch delay slot, to form a PC-relative effective target address.

If the contents of GPR *Rs* and GPR *Rt* are not equal, branch to the effective target address after the instruction in the delay slot is executed.

Restrictions:

The *Offset* value must be an integer in the range [0, 1]. Using an *Offset* value outside of this range leads to undefined behavior.

Operation:

```
if (GPR[Rs] == GPR[Rt]) then
    PC += Offset + 2
else
    PC += 1
endif
```

Exceptions:

None

# PRINT

Print Signed Byte

7                      6 5                      4 3                      1 0                      0

SPECIAL 1 1	Rs s s	0 0 0 0	PRINT 1
----------------	-----------	------------	------------

2

2

3

1

Format:        PRINT Rs

Purpose:        To print 8-bit integers.

Description:    print Rs

                The 8-bit byte value in GPR *Rs* is printed to the console.

Restrictions:

                None

Operation:

```
if (OVERFLOW[Rs]) then
    console.log(IntegerOverflow)
else
    console.log(GPR[Rs])
endif
```

Exceptions:

                None