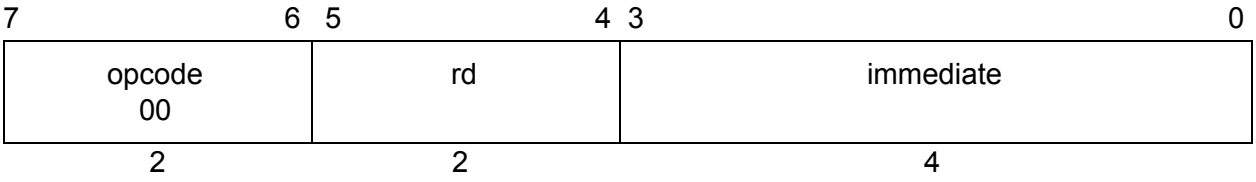
CPEG 324
LAB 1: Calculator ISA
Vinay Vazir & Jason Reynolds

ISA Design Document

INSTRUCTIONS

Mnemonic	Instruction
LOD	Load Immediate
ADD	Add Byte
SUB	Subtract Byte
DSP	Display Byte
CMP	Compare Bytes

Load Immediate **LOD**



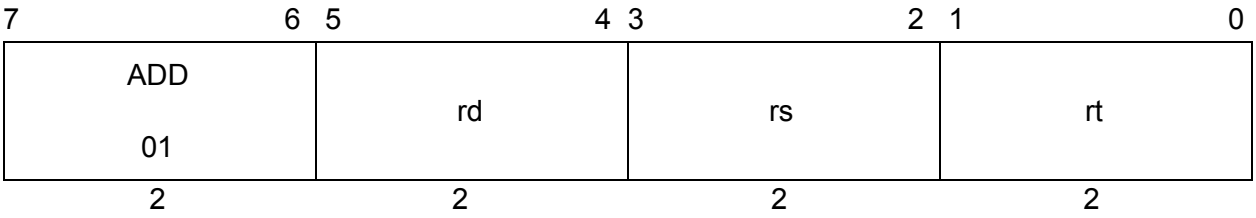
Format: `lod rd immediate`

Purpose:
To load a constant into the specified register.

Description:
The 4-bit signed *immediate* is loaded into the 8-bit general purpose register *rd*.

Programming Notes:
When using the *lod* command, the immediate needs to be entered as a 4-bit signed binary number, using two's complement to represent the negative numbers.

Add **ADD**



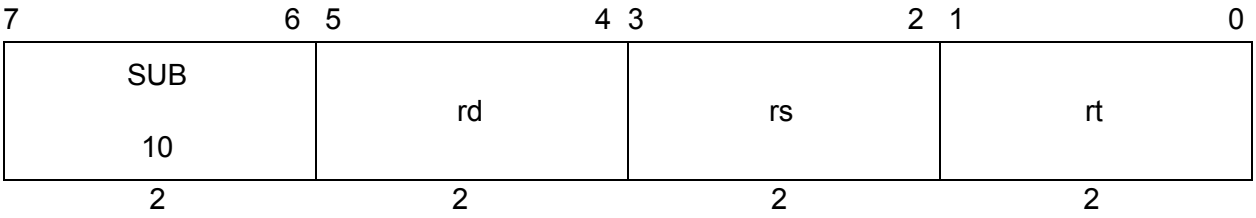
Format: `add rd rs rt`

Purpose:
To add values stored in registers.

Description:
The value stored in the 8-bit general purpose register *rt* is added to the value stored in the 8-bit general purpose register *rs* and stored in the 8-bit general purpose register *rd*. Overflows are ignored, sign is extended.

Programming Notes:
none

Sub **SUB**



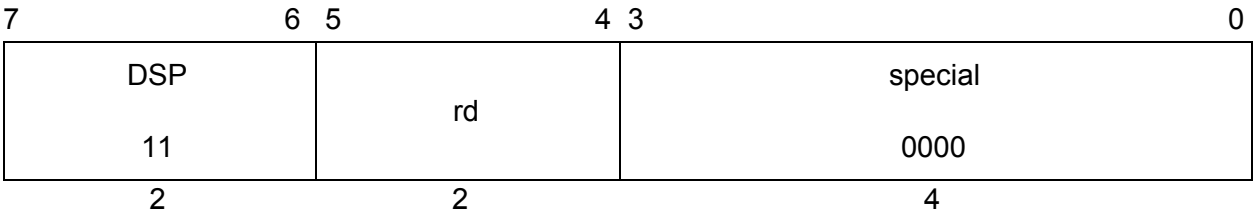
Format: `sub rd rs rt`

Purpose:
To subtract the values stored in the registers.

Description:
The 8-bit signed integer value stored in the general purpose register *rt* is subtracted from the 8-bit signed integer value stored in the general purpose register *rs* and stored in the general purpose register *rd*.

Programming Notes:
none.

Display **DSP**



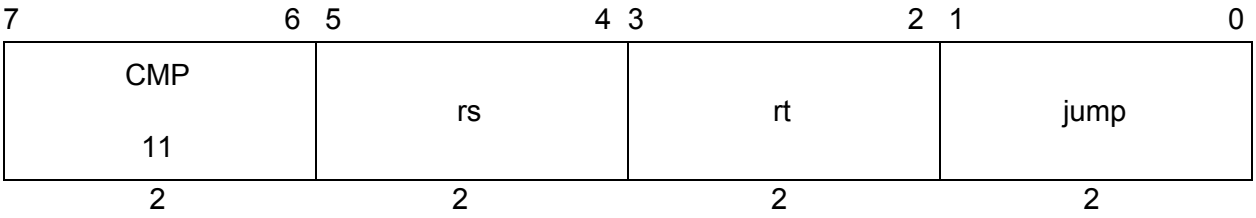
Format: `dsp rd`

Purpose:
To display values stored in the registers.

Description:
The 8-bit signed integer value stored in the general purpose register *rd* is displayed on standard out in both decimal and binary format.

Programming Notes:
none.

Compare **CMP**



Format: `cmp rs rt jump`

Purpose:
To compare the values stored in the general purpose registers *rs* and *rt* and either continue to the next line, or skip one or two lines of code depending on the value of *jump*.

Description:
The 8-bit signed integer value stored in the general purpose register *rs* is compared to the 8-bit signed integer value stored in the general purpose register *rt*. In the case that they are of equal value, the program counter is advanced by either an additional 1, or an additional 2 increments, thereby skipping either the line immediately after this instruction, or the two lines immediately following this instruction. In the case that the the values in *rs* and *rt* are not equal, the program continues on to the next line uninterrupted.

Programming Notes:
The value for *jump* must be entered in decimal format and equal to either “1” or “2”.