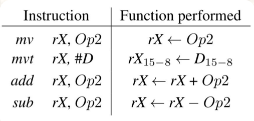

Instruction		Function performed
mv	*rX*, *Op*2	$rX \leftarrow Op2$
mvt	*rX*, #*D*	$rX_{15-8} \leftarrow D_{15-8}$
add	*rX*, *Op*2	$rX \leftarrow rX + Op2$
sub	*rX*, *Op*2	$rX \leftarrow rX - Op2$

*Op2 can be either: ① constant, expressed as #D, limit to 8 bits in size

② Another register rY.

*mvt: copies the 8-bit constant into the most significant bits of rx, i.e: $rX_{15\text{-}8}$ & set $rX_{7\text{-}0} = \emptyset$

- How to encode these 4 instructions into 16 bit machine code? $C_{15}C_{14} \cdots C_2C_1C_0$

There are two cases to consider for these "4" code types:

1. If Op2 is a constant. D ("immediate Data") encoding has this form: $\underbrace{III}_{\text{instruction type}}\ \overset{\text{constant}}{1}\ \underbrace{\overset{\text{ignore}}{XXX}}_{rX}\ \underbrace{DDDDDDDD}_{\text{8-bit constant}}$

2. Op2 is a register rY. Y is one of 0→7, 16 bits encoding is III 0 XXX $000000\underbrace{YYY}_{rY}$

III	instruction
000	mv
001	mvt
010	add
011	sub

example:

mv r1, r5 $r_1 \leftarrow r_5$: 000 0 001 000000 101

add r7, r6 $r_7 \leftarrow r_7 + r_6$: 010 0 111 000000 110

add r3, #8 $r_3 \leftarrow r_3 + 8_{10}$: 010 1 011 000001000 (base 16: 0x1A)

example for terminology: (human readable/understandable form of instruction)

Assembly instruction	Machine Code / Object code	(what computer understand)	
mv r2, #243	000 1 010 $0\ \overset{243}{\underline{11110011}}$	$r_2 \leftarrow 243$	
mvt r1, #0xFF00	001 1 001 $0\ \overset{FF}{\underline{11111111}}$	$r_1 \leftarrow FF00_{16}$	} after this $r_1 = FFFF_{16}$ (all 1's)
add r1, #0xFF	010 1 001 0 11111111	$r_1 \leftarrow r_1 + FF_{16}$	
sub r1, r2	011 0 $\overset{r_1}{\underline{001}}$ $0\ 00000\overset{r_2}{\underline{010}}$	$r_1 \leftarrow r_1 - r_2$	
add r1, #1	010 1 001 0 00000001	$r_1 \leftarrow r_1 + 1$	