ARM Conditional Branch Instructions

Description	Symbol	Java	Instruction	Mnemonic
Equality				
equal	=	==	BEQ	**EQ**ual
not equal	≠	!=	BNE	**N**ot **E**qual
Inequality (unsigned values)				
less than	<	<	BLO (or BCC)	**LO**wer
less than or equal	≤	<=	BLS	**L**ower or **S**ame
greater than or equal	≥	>=	BHS (or BCS)	**H**igher or **S**ame
greater than	>	>	BHI	**HI**gher
Inequality (signed values)				
less than	<	<	BLT	**L**ess **T**han
less than or equal	≤	<=	BLE	**L**ess than or **E**qual
greater than or equal	≥	>=	BGE	**G**reater than or **E**qual
greater than	>	>	BGT	**G**reater **T**han
Flags				
Negative Set			BMI	**MI**nus
Negative Clear			BPL	**PL**us
Carry Set			BCS (or BHS)	**C**arry **S**et
Carry Clear			BCC (or BLO)	**C**arry **C**lear
Overflow Set			BVS	o**V**erflow **S**et
Overflow Clear			BVC	o**V**erflow **C**lear
Zero Set			BEQ	**EQ**ual
Zero Clear			BNE	**N**ot **E**qual

Equality and Inequality Mnemonics are based on a previous execution of a compare (`CMP`) instruction of the form `CMP Rx, Ry`. For example, `BLE label` will branch to `label` if *Rx* is less than or equal to *Ry*.

Pseudo Code Examples

Pseudo Code	ARM Assembly Language

`if (x <= y)`
`{`
`  x = x + 1;`
`}`
 	
assume x and y are signed values
	`CMP  Rx, Ry`
	`BGT  label`
	`ADD  Rx, Rx, #1`
	`label`

`if (x < y)`
`{`
`  z = x;`
`}`
`else {`
`  z = y;`
`}`
 	
assume x and y are unsigned values
	`CMP  Rx, Ry`
	`BHS  label1`
	`MOV  Rz, Rx`
	`B  label2`
	`label1`
	`MOV  Rz, Ry`
	`label2`

`while (x > 2)`
`{`
`  y = x * y;`
`  x = x - 1;`
`}`
 	
assume x and y are unsigned values
	`label1  CMP  Rx, #2`
	`BLS  label2`
	`MUL  Ry, Rx, Ry`
	`SUB  Rx, Rx, #1`
	`B  label1`
	`label2`