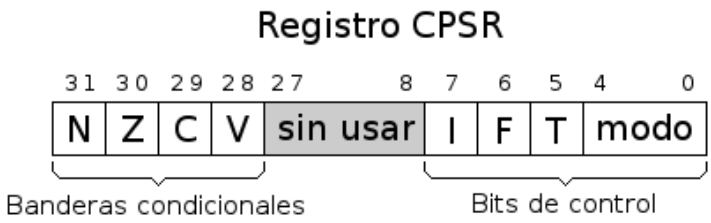
Organización del Computador UNGS

Set de Instrucciones ARM

S: Es un sufijo opcional. Si se especifica S, los flags se actualizan en el resultado de la operación. **cond:** Es un código de condición opcional. **rd:** Es el registro de destino **Rn**: Es el registro que contiene el primer operando. **Op2** : segundo operando. Podría ser imn. **imm**: Inmediato, es cualquier valor en el rango de 0-4095. **label:** Expresión relativa a PC. **Rm:** Es un registro que contiene una dirección de salto

Mnemónico	Instrucción	{S} Flags	Acción
ADC {S}{cond} {Rd}, Rn, Op2	Suma con carry	**N Z C V**	Rd:= Rn + Op2 + Carry
ADD {S} {cond} {Rd}, Rn, Op2 **ADD {S} {cond} {Rd}, Rn, imm**	Suma	**N Z C V**	Rd := Rn + Op2
AND {S} {cond} Rd, Rn, Op2	Y lógico	**N Z C**	Rd := Rn AND Op2
B {cond} label	Salto	-	R15 := address (±32MB)
BAL label	Salto incondicional	-	R15 := address (±32MB)
BIC {S}{cond} Rd, Rn, Operand2	Bit Clear	**N Z C**	Rd := Rn AND NOT Op2
BL {cond} label	Branch with Link	-	R14 := R15, R15 := address (±32MB)
BX {cond} Rm	Branch and Exchange	-	R15 := Rn, T bit := Rn[0]
CDP {cond} coproc, #opcode1, CRd, CRn, CRm{, #opcode2}	Coprocessor Data Processing	-	(Coprocessor-specific)
CMN {cond} Rn, Operand2	Compara Negativo	**N Z C V**	CPSR flags := Rn + Op2
CMP {cond} Rn, Operand2	Compara	**N Z C V**	CPSR flags := Rn - Op2
EOR {S}{cond} Rd, Rn, Operand2	OR Exclusivo		Rd := (Rn AND NOT Op2) OR (op2 AND NOT Rn)
LDR Rd, [Rn]	Carga registro desde memoria	-	Rd := [Rn] Carga en Rd, el valor que se encuentra en la dirección [Rn], Rn contiene una dirección
LDM	Load Multiple registers.	-	Stack manipulation (Pop) LDM r8,{r0,r2,r9}
MLA {S}{cond} Rd, Rn, Rm, Ra	Multiplica y acumula	**N Z**	Rd := (Rm * Rn) + Ra **Rn, Rm** son registros que almacenan los valores a multiplicar. **Ra** tiene un valor que se suma al producto obtenido

MOV {S}{cond} Rd, Op2 **MOV {cond} Rd, #imm**	Move register or constant	**N Z C**	Rd : = Op2 Rd:= imm
MUL {S}{cond} {Rd}, Rn, Rm	Multiply with signed or unsigned 32-bit operands, giving the least significant 32 bits of the result.	**N Z**	Rd := Rn * Rm **Rd:** registro destino. **Rn, Rm**: contienen los valores a multiplicar
MVN {S}{cond} Rd, Op2	Move negative register	**N Z C**	Rd := 0xFFFFFFFF EOR Op2
NEG {cond} Rd, Rm	Negate the value in a register.		Pseudo-instrucción actualiza los flags basado en el resultado
NOP {cond}	No Opera	-	**NOP** No necesariamente consume tiempo. El procesador puede quitarlo del pipeline antes de ejecutar.
ORR {S}{cond} Rd, Rn, Op2	Logical OR	**N Z C**	Rd := Rn OR Op2
RSB {S}{cond} {Rd}, Rn, Op2	Reverse Subtract without carry.	**N Z C V**	Rd := Op2 - Rn ejemplo: RSB r4, r4, #1280
ROR {cond} Rd, Rm, Rs **ROR {cond} Rd, Rm, #sh**	Rotate Right.		Desplaza hacia la derecha, los bits contenidos en el registro **sh:** constante con rango a desplazar 1-31, también se puede usar Rs para este valor
RSC {S}{cond} {Rd}, Rn, Op2	Reverse Subtract with Carry	**N Z C V**	Rd := Op2 - Rn - 1 + Carry Si el carry está limpio se le resta 1 al resultado.
SBC {S}{cond} {Rd}, Rn, Op2	Subtract with Carry	**N Z C V**	Rd := Rn - Op2 - 1 + Carry
SUB {S}{cond} {Rd}, Rn, Op2	Subtract	**N Z C V**	Rd := Rn - Op2
STM	Store Multiple	-	Stack manipulation (Push)
STR Rd, [Rb]	Guarda registro a memoria	-	<address> := Rd Guarda el valor contenido en Rd en la dirección de memoria apuntada por Rb Rb, contiene una dirección de memoria
SWI #imm	Software Interrupt	-	OS call imm: 0-255
SWP {B}{cond} Rd, Rn, [Rm]	Swap register with memory	-	Rd := [Rn], [Rn] := Rm
TST {cond} Rn, Op2	Test bits		CPSR flags := Rn AND Op2
PUSH `{cond} reglist`	Push registers onto a full descending stack.	-	`PUSH {r0,r4-r7}` `PUSH {r2,lr}`
POP {cond} reglist	Pop registers off a full descending stack	-	POP {r0,r10,pc}



Condition Code

Code	Suffix	Flags	Meaning
0000	EQ	Z set	Equal
0001	NE	Z clear	Not equal
0010	CS	C set	Unsigned higher o same
0011	CC	C clear	Unsigned lower
0100	MI	N set	Negative
0101	PL	N Clear	Positve or zero
0110	VS	V set	Overflow
0111	VC	V clear	No overflow
1000	HI	C set and Z clear	Unsigned higher
1001	LS	C clear or Z set	Unsigned lower or same
1010	GE	N equals V	Greater o equal
1011	LT	N not equal to V	Less tan
1100	GT	Z clear AND (N equal to V)	Greater tan
1101	LE	Z set OR (N not equal to V)	Less tan or equal
1110	AL	(ignored)	always