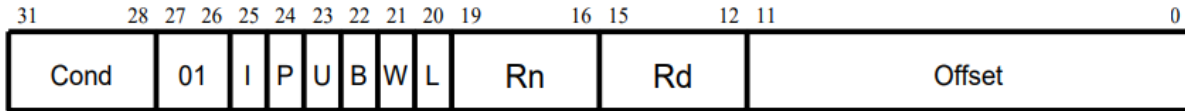
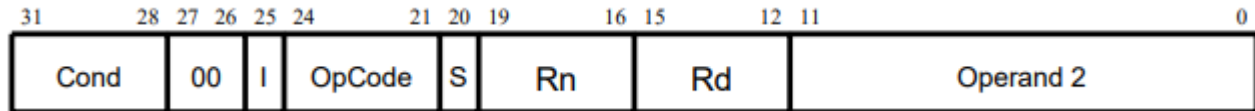


Single Data Transfer



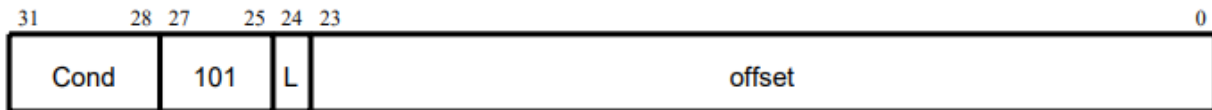
Assembly Language Symbol	Instruction
LDR	Load register from memory
STR	Store register to memory

Data Processing



Assembly Language Symbol	Instruction	OpCode
MOV	Move register or constant	1101
ADD	Addition of operands from registers	0000
CMP	Compare values in register	1010
SUB	Subtraction of operands from registers	0010
ORR	Bitwise OR operation	1100
ADC	Addition of operands with carry	0101

Branch



Assembly Language Symbol	Instruction
BLT	Branch if less than

Assembly Program

```
MOV R4, #0           ; register to store the sum
MOV R0, #0           ; index i, will be used in the loop
MOV R3, #16          ; final size of the array, 4 words
LDR R1, =A           ; load array A into register 1
LDR R2, =B           ; load array B into register 2
LDR R7, =C           ; load array C into register 5, it will be the empty array where results will be saved
LOOP:
    LDR R5, [R1], #4  ; load value of array A into register 5 and update base register by R1 = R1 + 4
    LDR R6, [R2], #4  ; load value of array A into register 6 and update base register by R2 = R2 + 4
    ADD R4, R5, R6     ; add values from register 5 and 6 and place the result in register 4
    STR R4, [R7], #4   ; store value into R7 and update base register by R7 = R7 + 4
    ADD R0, #4         ; increase counter by 1 word = 4 bytes
    CMP R0, R3         ; check if we are done
    BLT LOOP          ; if not done loop again
```