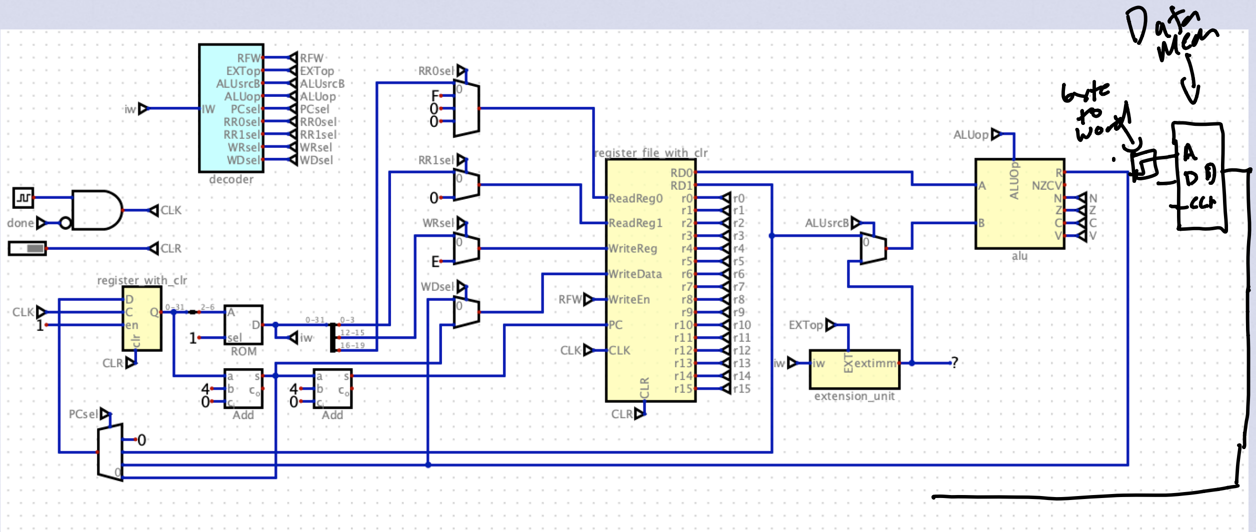
NZCV bits

Data Memory and stack setup
ldr / str

Decoder → Control Unit → add conditional execution

lsr / lsl ← mov

Data Memory



stack initialization and use

main:
mov sp, #128
sub sp, sp, #4
mov r0, #99
str r0 [sp]

128

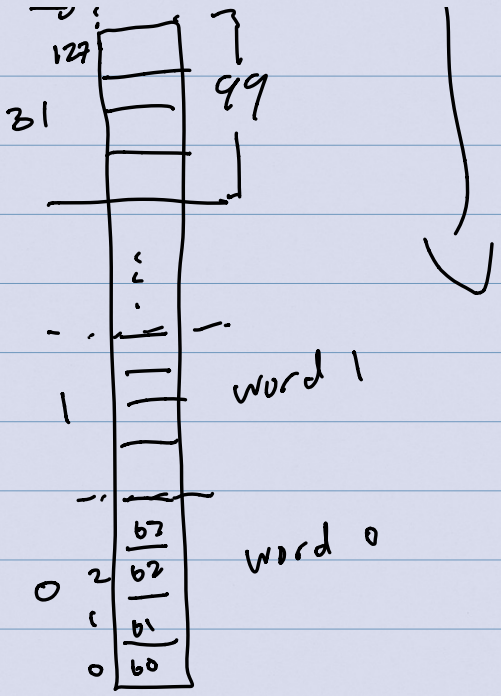
5 addr bits
2^5 words
32 word
32×4 = 128 bytes

```
int array[] = {1, 2, 3, 4};

main:
    mov sp, #128
    sub sp, sp, #16
    mov r0, #1
    str r0, [sp, #0]
    mov r0, #2
    str r0, [sp, #4]
    mov r0, #3
    str r0, [sp, #8]
    mov r0, #4
    str r0, [sp, #12]

    mov r0, sp
    mov r1, #4
    bl sum_array_s
    add r0, r0, #0

sum_array_s:
    .
    .
    .
```

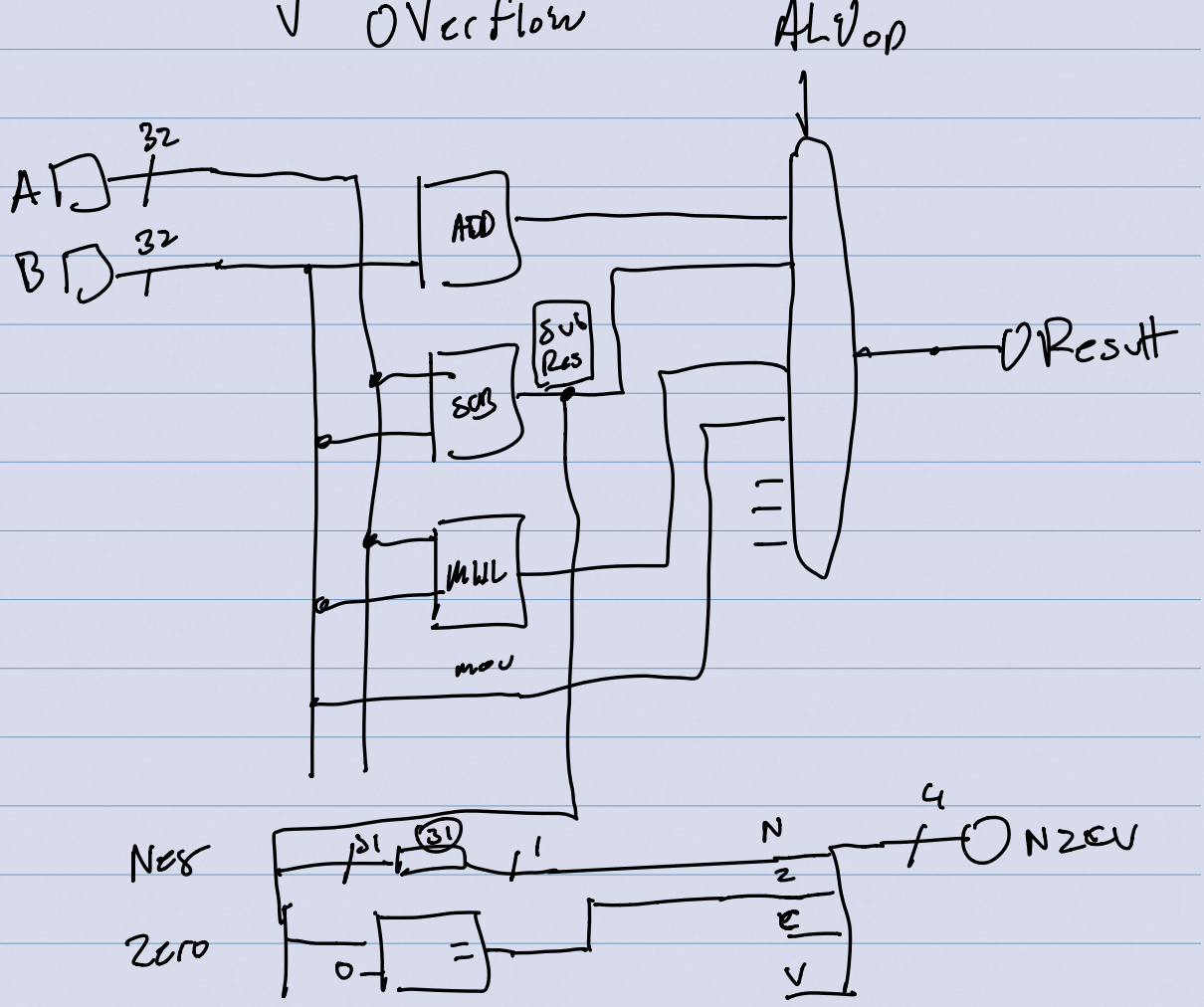



```
mov sp, #256
    7 addr bits?
2^7 = 128 words
    128 x 4 = 512 bytes

mov sp, #128
add sp, sp, sp   256
add sp, sp, sp   512
mov r0, #4
mul r1, r0, sp   512
```

ALU NZCV

N negative
Z zero
C carry
V Overflow



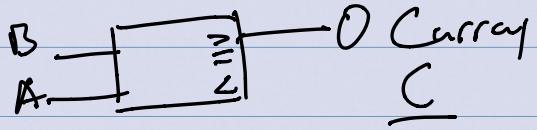
Carry unsigned addition

$A > 2^{31}$
$B > 2^{31}$
$\Rightarrow A+B > 2^{32}$

unsigned subtraction

A - B if B > A then need a borrow

↑ Carry



Overflow signed

Addition

+A + +B	< 0	overflow
-A + -B	> 0	overflow
+A + -B		no overflow
-A + +B		no overflow

Subtraction		Result	
⇒ (1)	+A - -B	< 0	overflow
(2)	-A - +B	> 0	overflow
	+A - +B		no overflow
	-A - -B		no overflow