

NZCV Bits

Extender for branch 24 bit sign extend and  $\times 4$   
can use MUL component or shifter

Processor components

Register File

ALU (NZCV)

Extender

Instruction Memory

Data Memory (stack) How to structure main  
in tests

Decoder → Control Unit

PC

↳ Conditional  
Execution  
CPSR

MUXes

Adders

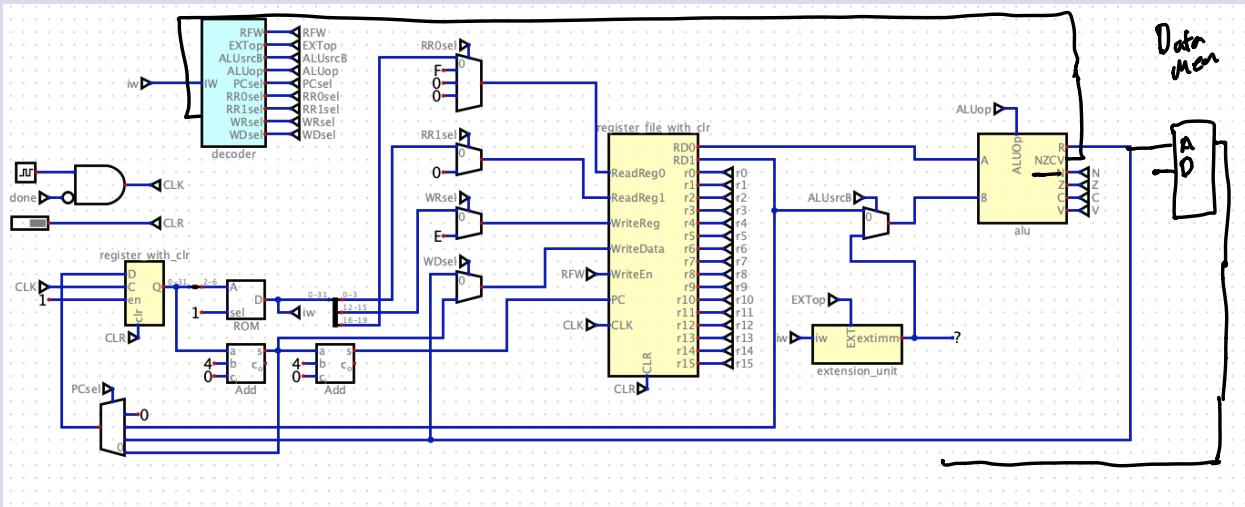
Data Path

Control Path

Decoder / Control Unit Output

to Components

lsr / lsl ← mov



Stack initialization and osc

Main:

MOV SP, #128

SUB SP, SP, #4

MOV R0, #9

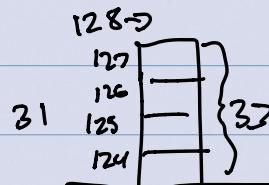
STR R0, [SP]

5 addr bits

$2^5$  words

32 words

$$32 \times 4 = 128 \text{ bytes}$$



Data Memory

Main:

MOV SP, #128

SUB SP, SP, #16

MOV R0, #1

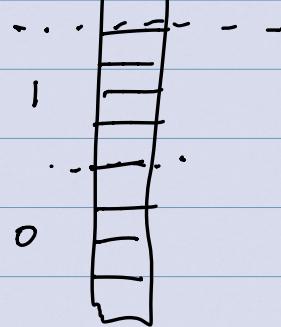
STR SP, [SP, #0]

MOV R0, #2

STR SP, [SP, #4]

MOV R0, #3

STR SP, [SP, #8]



$$\text{array[7] = \{ 1, 2, 3, 4 \}}$$

```

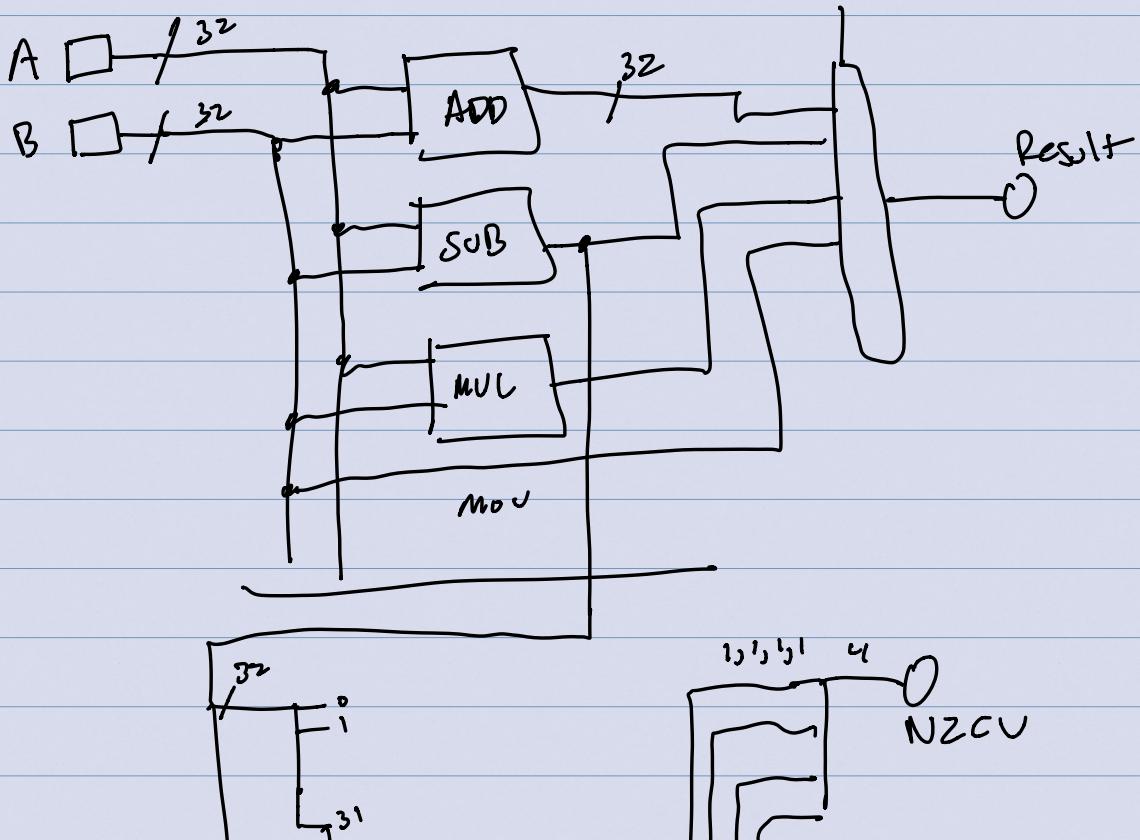
    mov r0, #4
    str sp, [sp, #12]
    mov r0, sp
    mov r1, #4
    b1 sum_array_s
    add r0, r0, #0
sum_array_s:
;
;
;

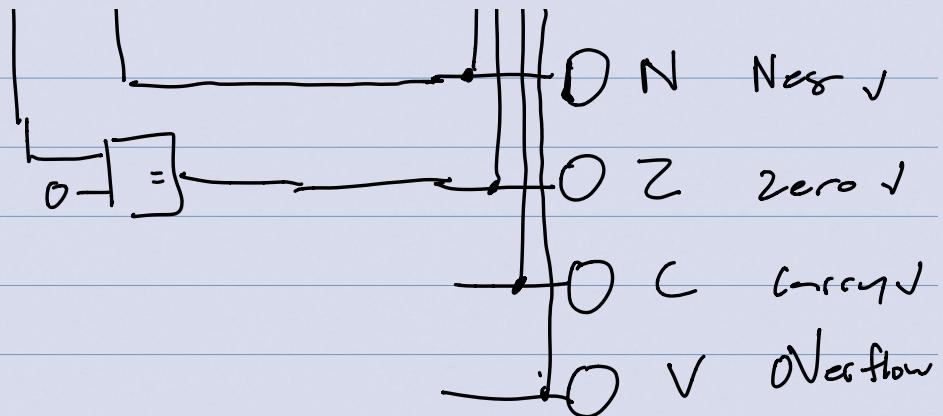
```

~~mov r0, #255~~ To handle large stack sizes  
 mov r0, #128  
 add r0, r0, r0 // r0 = 256  
 add r0, r0, r0 // r0 = 512  
;  
 mov r0, #128  
 mov r1, #4
 mul r2, r1, r0
 // r2 = 512

ALU N2CV

ALUop





carry      32 bits

→ unsigned  
subtraction

$$A > 2^{31}$$

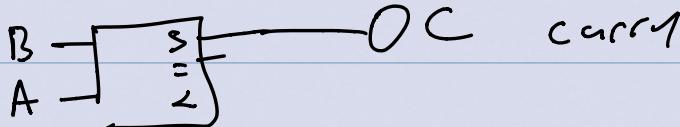
$$B > 2^{31}$$

$$A + B > 2^{23} \quad X$$

For sub

$$\text{if } B > A$$

$$A - B \Rightarrow \text{carry}$$



Overflow signed

Addition

$$+A + +B < 0 \quad \text{overflow}$$

$$-A + -B > 0 \quad \text{overflow}$$

$$+A + -B \quad \text{no overflow}$$

$$-A + +B \quad \text{no overflow}$$

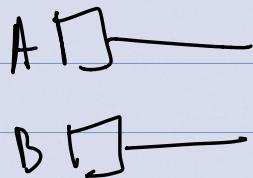
## Subtraction

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

is A neg?  $A[31]$

is B neg?  $B[31]$

is R neg?  $R[31]$



$\overline{R}$

Overflow

