- √ Test program construction stack / arrays
- √ Spreadsheet construction
- √ SDT instructions → MR and MW
- Conditional execution N Z C V + CPSR + CPSRW

Test programs:

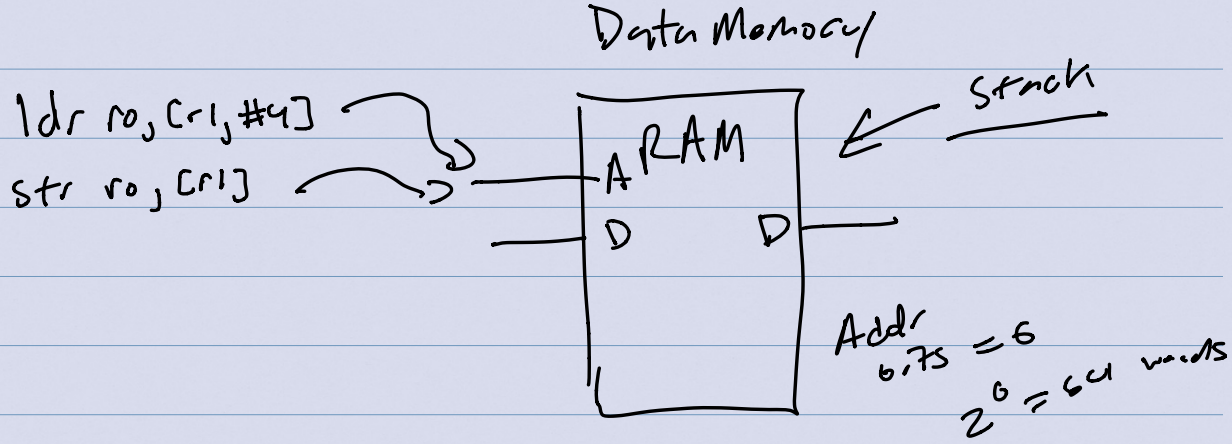
sum_array_main.c sum_array_s.s → .hex

```
main() {
    int a[] = {1,2,3};
    int r;
    :
    r = sum_array_s(a,3)
    printf("r=%d\n", r);
    exit(0)
}
```

```
sum_array_s:
    :
    :
    :
    bx lr
```

sum_array_main.s → armasm → sum_array_main.hex

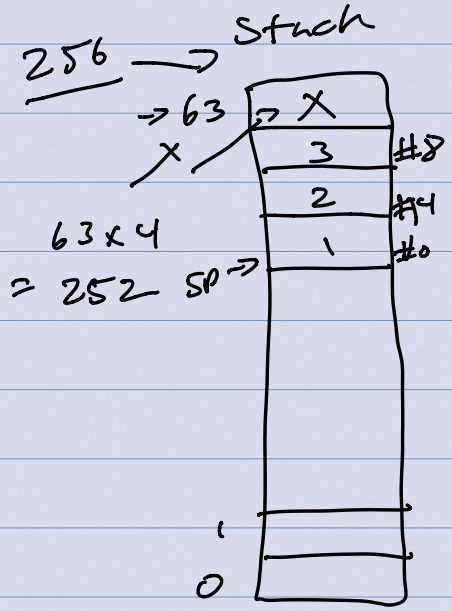
main:



sum_array_main.s → armasm → sum_array_main_hex

```
main:
    mov sp, #128
    add sp, sp, sp
    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, sp
    mov r1, #3
    bl  sum_array_s
    add r0, r0, #0
    :

sum_array_s:
    :
    bx lr
```



$2^6 = 64$

64 x 4 = 256 bytes

0 to 255

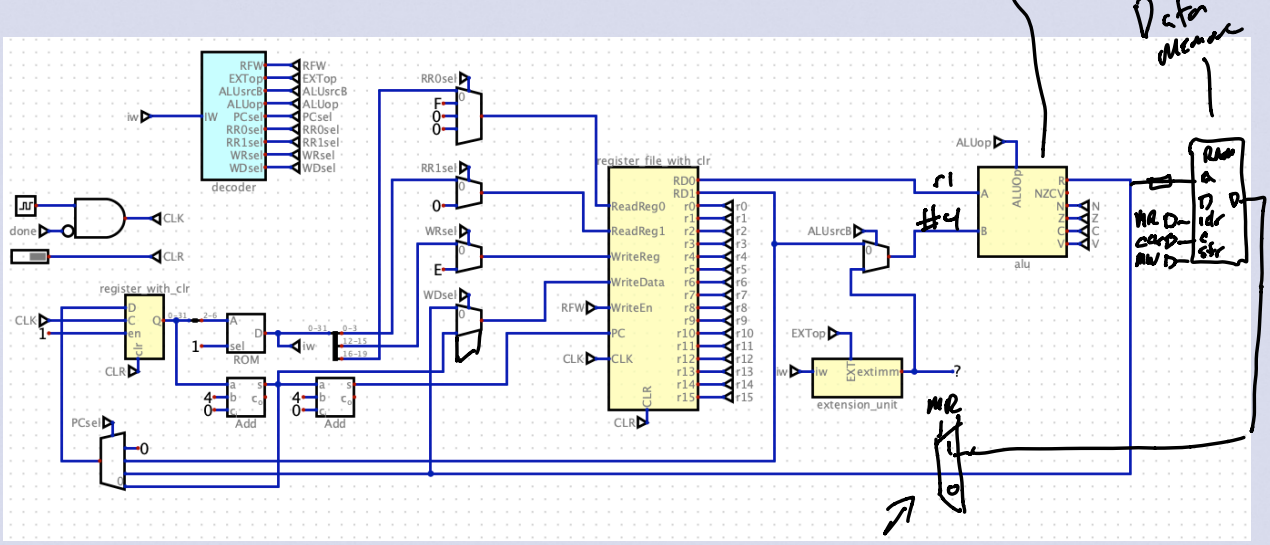
Incremental Work

Data Processing: mov(r), add(i)
sub(r), sub(i)
cmp(r), cmp(i) →
lsl(r), lsl(i) ⇒ mov
lsr(r), lsr(i)

Multiply: mul
SDT: ldr, str
Cond execution for branch

SDT ldr/str

ldr r0, [r1, #4]



ldr r0, [r1]
ldr r0, [r1, #0]
ldr r0, [r1, r2]

str r0, [r1]
str r0, [r1, #0]
~~str r0, [r1, r2]~~

Conditional Execution

```
cmp r0, #0
beq end
```

step 1

cmp

Subtract in ALU
No WriteEn for RegFile

NCZV

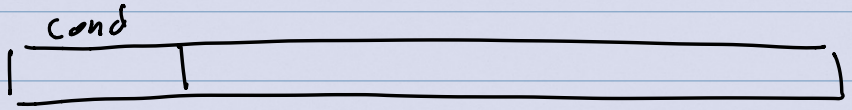
In control unit add CPSR Reg (4 bits)

step 2

b(*) beq, bne, blt, bgc, ...

Act on cond code Field

IV



E

1110