Module 3

2.3] B[8] = A[i-j] →

A is in $s6
B is in $s7

```
sub  $t0, $s3, $s4   # i - j
add  $t0, $t0, $s6   # add the base to $t0
lw   $t0, 0($t0)     # load A[8] into $t0
sw   $t0, 32($s7)    # load $t0 into B[8]
```

2.4]

```
sll  $t0, $s0, 2     # t0 = 0x4
add  $t0, $s6, $t0   # t0 = &A[f]   as $s6 has the base register
sll  $t1, $s1, 2     # $t1 = g x 4  $shift let $s1 by 2 which = a multiplication by 4
add  $t1, $s7, $t1   # $t1 = &B[g]  $s7 the base of B
lw   $s0, 0($t0)     # f = A[f]
```

in

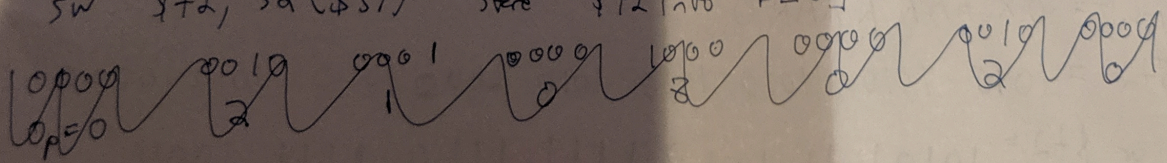
B[g] = A[f] + A[f+1]

sw $t0, 0($t1) store $t0 in $t1 where $t1 is the memory address of B[g]

2.9] i is in $s3 so lets find the address of A[i] first

```
sll  $t0, $s3, 2     # $t0 = i * 4
add  $t0, $t0, $s6   # add the base addr of A
lw   $t0, 0($t0)     # load value of A[i] into $t0
sll  $t1, $s4, 2     # same for j. j is in $s4
add  $t1, $t1, $s6   # add base of A
lw   $t1, 0($t1)     # load A[j] into $t1
add  $t2, $t0, $t1   # add both values and store into $t2
sw   $t2, 32($s7)    # store $t2 into B[8]
```

2.14]



000000	10000	10000	10000	00000	100000
op=0	rs=16	rs=16	rs=16	shamt=0	funct=0x20 (add)

add $s0, $s0, $s0 R-type

2.15] sw $t1, 32($t2)

43 | 10 | 9 | 32
101011 | 01010 | 01001 | 0000 0000 0010 0000

opcode
sw
43

I-type 0xAD490020

2.16] op=0 rs=3 rt=2 rd=3 shamt=0 funct=34

sub because of the 0 and 34

pg. 105
rs=3 ; $v1
rt=2 ; $v0
rd=3 ; $v1

R-type
sub $v1, $v1, $v0

2.17] op=0x23 , rs=1 , rt=2, const=0x4

hex=35 | $at | $v0

I-type
lw $v0, 4($at)

100 0011 | 00001 | 00010 | 0000000000000100

35 | 1 | 2 | 4
0x23

2.19] $t0 = 0xAAAAAAAA $t1 = 0x12345678

2.19.1] $t0 in bytes = 1010 1010 1010 1010 1010 1010 1010 1010
$t1 in bytes = 0001 0010 0011 0100 0101 0110 0111 1000

sll result = 0xAAAAAAA0

$t2 = 1010 1010 1010 1010 1010 1010 1010 0000

$t1 or $t2 = 10101011 1110 1111 1110 1111 1000

1011

0x BA BE FE F8

2.19.2)

$t2 1010 1010 1010 1010 1010 1010 1010 0000

so $t2 = 0xAAAA AAA0

andi $t2, $t2, -1

-1 = 1111 1111 1111 1111 1111 1111 1111 1111

$t2 after andi = 1010 1010 1010 1010 1010 1010 1010 0000

= 0xAAAA AAA0

2.19.3)

srl $t2, $t0, 3

$t2 0001 0101 0101 0101 0101 0101 0101 0101

0x15555555

andi $t2, $t2, 0xFFEF

0xFFEF = 0000 0000 0000 0000 1111 1111 1110 1111

andi $t2, $t2 = 0000 0000 0000 0000 0101 0101 0100 0101

0xFFEF = 0x00005545