Nick Petro (collaborated with Kevin Gilboy)

1/22/16

COE 0147 - Lab 5

Book Problems

2.31)

fib:

addi $sp, $sp, -12 # allocate stack frame of 12 bytes

sw $a0, 8($sp) # save n

sw $ra, 4($sp) # save return address

sw $s0, 0($sp) # save $s0

slti $t0, $a0, 2 # fib(i) = i for i = 0, 1

beq $t0, $0, else

add $v0, $a0, $0 # $v0 = 0 or 1

j exit # go to exit

else:

addi $a0, $a0, -1 # fib(n-1)

jal fib # recursive call

add $s0, $v0, $0

addi $a0, $a0, -1 # fib(n-2)

jal fib # recursive call

add $v0, $v0, $s0

exit:

lw $a0, 8($sp) # restore $a0

lw $ra, 4($sp) # restore return address

lw $s0, 0($sp) # restore $s0

addi $sp, $sp, 12 # free stack frame

jr $ra # return to caller

2.34)

f: addi $sp, $sp, -4 # allocate stack frame of 4 bytes

sw $ra, 0($sp) # save return address

jal func # call func(a,b)

add $a0, $v0, $0 # $a0 = result of func(a,b)

add $a1, $a2, $a3 # $a1 = c+d

jal func # call func(func(a,b), c+d)

lw $ra, 0($sp) # restore return address

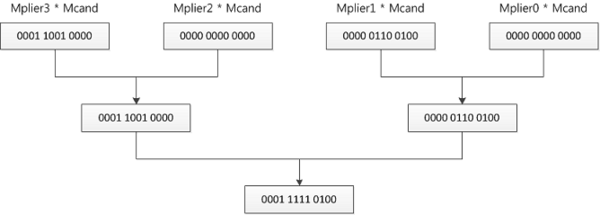
addi $sp, $sp, 4 # free stack frame

jr $ra # return to caller

1)

|  |  |  |
| --- | --- | --- |
| 144-28 10010000 – 00011100  10010000 + (11100011+1)  10010000 + 11100100    Since leading 0, the number is positive. | 92-43 1011100 – 0101011  1011100 + (1010100+1)  1011100 + 1010101  Since leading 0, the number is positive. | -32-84 -0100000 – 1010100  (1011111+1) + (0101011+1)  1100000 + 0101100  Since leading 1, the number is negative, must take the complement.  -(01110011-1)  -01110010 |

3.12)



062 000 110 010

\* 012 \* 000 001 010

764 111 110 100

3.13)

062 0000 0110 0010

\* 012 \* 0000 0001 0010

6E4 0110 1110 0100

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration** | **Step** | **Multiplier**  **(lower 32bits)** | **Multiplicand** | **Product**  **(upper 32bits)** |
| 0 | initial values | 0001 0010 | 0000 0110 0010 | 0000 0000 | 0000 0000 |
| 1 | 1: 0 ⇒ No operation  2: Shift left Multiplicand  3: Shift right Product | 0001 0010  0001 0010  0000 1001 | 0000 0110 0010  0000 1100 0100  0000 1100 0100 | 0000 0000 | 0000 0000  0000 0000 | 0000 0000  0000 0000 | 0000 0000 |
| 2 | 1a: 1 ⇒ Prod = Prod+Mcand  2: Shift left Multiplicand  3: Shift right Product | 0000 1001  0000 1001  0000 0100 | 0000 1100 0100  0001 1000 1000  0001 1000 1000 | 0110 0010 | 0000 0000  0110 0010 | 0000 0000  0011 0001 | 0000 0000 |
| 3 | 1: 0 ⇒ No operation  2: Shift left Multiplicand  3: Shift right Product | 0000 0100  0000 0100  0000 0010 | 0001 1000 1000  0011 0001 0000  0011 0001 0000 | 0011 0001 | 0000 0000  0011 0001 | 0000 0000  0001 1000 | 1000 0000 |
| 4 | 1: 0 ⇒ No operation  2: Shift left Multiplicand  3: Shift right Product | 0000 0010  0000 0001  0000 0001 | 0011 0001 0000  0110 0010 0000  0110 0010 0000 | 0001 1000 | 1000 0000  0001 1000 | 1000 0000  0000 1100 | 0100 0000 |
| 5 | 1a: 1 ⇒ Prod = Prod+Mcand  2: Shift left Multiplicand  3: Shift right Product | 0000 0001  0000 0001  0000 0000 | 0110 0010 0000  1100 0100 0000  1100 0100 0000 | 0110 1110 | 0100 0000  0110 1110 | 0100 0000  0011 0111 | 0010 0000 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteration** | **Step** | **Quotient** | **Divisor** | **Remainder** |
| 0 | initial values | 000000 | 010001 000000 | 000000 111100 |
| 1 | 1: Rem = Rem - Div  2b: Rem < 0 ⇒ +Div, sll Q, Q0=0  3: Shift Div right | 000000  000000  000000 | 010001 000000  010001 000000  001000 100000 | 101111 111100  000000 111100  000000 111100 |
| 2 | 1: Rem = Rem - Div  2b: Rem < 0 ⇒ +Div, sll Q, Q0=0  3: Shift Div right | 000000  000000  000000 | 001000 100000  001000 100000  000100 010000 | 111000 011100  000000 111100  000000 111100 |
| 3 | 1: Rem = Rem - Div  2b: Rem < 0 ⇒ +Div, sll Q, Q0=0  3: Shift Div right | 000000  000000  000000 | 000100 010000  000100 010000  000010 001000 | 111100 101100  000000 111100  000000 111100 |
| 4 | 1: Rem = Rem - Div  2b: Rem < 0 ⇒ +Div, sll Q, Q0=0  3: Shift Div right | 000000  000000  000000 | 000010 001000  000010 001000  000001 000100 | 111110 110100  000000 111100  000000 111100 |
| 5 | 1: Rem = Rem - Div  2b: Rem < 0 ⇒ +Div, sll Q, Q0=0  3: Shift Div right | 000000  000000  000000 | 000001 000100  000001 000100  000000 100010 | 111111 111000  000000 111100  000000 111100 |
| 6 | 1: Rem = Rem - Div  2a: Rem ≥ 0 ⇒ sll Q, Q0=1  3: Shift Div right | 000000  000001  000001 | 000000 100010  000000 100010  000000 010001 | 000000 011010  000000 011010  000000 011010 |
| 7 | 1: Rem = Rem - Div  2a: Rem ≥ 0 ⇒ sll Q, Q0=1  3: Shift Div right | 000001  000011  000011 | 000000 010001  000000 010001  000000 001000 | 000000 001001  000000 001001  000000 001001 |

3.18)

3.20)

two's complement integer : 201,326,592

unsigned integer : 201.326.592

3.22)

1.0 x 2-103

00001100000000000000000000000000

3.23)

63.25 = 253 x 2-2 = 1.1111101 x 25

01000010011111010000000000000000

3.24)

0100000001001111101000000000000000000000000000000000000000000000