Abedallah Abedrabbah

Project Assembly Conversion

a = X19, b = X20, f = X21, i = X9, g = X10, X22 =Val[0], XZR = X31

ADDI X19, XZR, #10 //a = 10

ADDI X20, XZR, #6 //b = 6

ADDI X22, XZR, #0 //X22 = Val [0]

ADDI X23, XZR, #40 //X23 = 40 for condition checking

MUL X21, X19, X20 //f = a \* b

ADD X10, X20, X19, //g = a + b

AND X9, XZR, XZR //set i = 0

Loop SUB X13, X23, X9 //X13 = 40 – i

CBZ X13, EndLoop //Checking of i is equal to 40

ADD X11, X22, X9 //X11 = &Val[i]

ADD X12, X9, X21, //X12 = f + i

STUR X12, [X11, #0] //Val[i] = X12 = f + 1

ADDI X9, X9, #1 //i++

B Loop

EndLoop ADD X11, X22, X10 //X11 = Val[g]

LSL X13, X10, #2 //g \* 4

SUB X14, X13, X11 //if(val[g] > (g\*4)) part 1

LSR X14, X14, #63 // if(val[g] > (g\*4)) part 2

CBZ, X14, ELSE //if(val[g] > (g\*4)) part 3

SUB X21, X19, X20 //f = a -b

ADDI X11, X22, #2048 //X11 = 2048 (7SEG) + &0X00

STUR X21, [X11, #0] //printf(“value of f is: ‘%f\n’, f); //Display on 7-//segment displays

B ENDING //Exits if statement and goes to end

ELSE ADD X21, X20, X19 //f = a + b

ADD X11, X22, #2048 //X11 =7-SEGMENT ADDRESS

STUR X21, [X11, #0] //printf(“value of f is: ‘%f\n’, f); //Display on 7-segment displays

ENDING B ENDING //Sends processor to infinite loop to end program

**Binary Conversion:**

1. ADDI X19, XZR, #10 10010001000000000010101111110011
2. ADDI X20, XZR, #6 10010001000000000001101111110100
3. ADDI X22, XZR, #0 10010001000000000000001111110110
4. ADDI X23, XZR, #40 1001000100
5. MUL X21, X19, X20 10011011000100110111111010010101
6. ADD X10, X20, X19, 10001011000101000000001001101010
7. AND X9, XZR, XZR 10001010000111110000001111101001
8. Loop SUB X13, X23, X9 11001011000101110000000100101101
9. CBZ X13, EndLoop 10110100000000000000000011001101
10. ADD X11, X22, X9 10001011000101100000000100101011
11. ADD X12, X9, X21, 10001011000010010000001010101100
12. STUR X12, [X11, #0] 11111000000000000000000110001011
13. ADDI X9, X9, #1 10010001000000000000010100101001
14. B Loop 00010111111111111111111111111001
15. EndLoop ADD X11, X22, X10 10001011000101100000000101001011
16. LSL X13, X10, #2 11010011011010100000100000001101
17. SUB X14, X13, X11 11001011000011010000000101101110
18. LSR X14, X14, #31 1101001101000000
19. CBZ X14, ELSE 10110101000000000000000010101110‬‬
20. SUB X21, X20, X19 11001011000101000000001001110101
21. ADDI X11, X22, #2048 10010001001000000000001011001011
22. STUR X21, [X11, #0] 11111000000000000000001010101011
23. B ENDING 00010100000000000000000000000100
24. ELSE ADD X21, X20, X19 10001011000101000000001001110101
25. ADDI X11, X22, #2048 10010001001000000000001011001011
26. STUR X21, [X11, #0] 11111000000000000000001010101011
27. ENDING B ENDING 00010100000000000000000000000000

|  |  |  |  |
| --- | --- | --- | --- |
| 0: | 8: | 16: | 24: |
| 1: | 9: | 17: | 25: |
| 2: | 10: | 18: | 26: |
| 3: | 11: | 19: |  |
| 4: | 12: 91000529 | 20: |  |
| 5: | 13: | 21: |  |
| 6: | 14: | 22: |  |
| 7: | 15: | 23: |  |