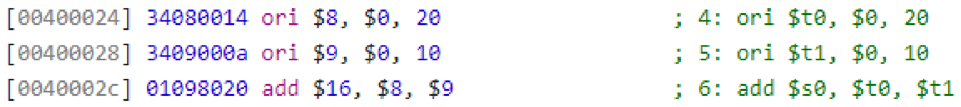
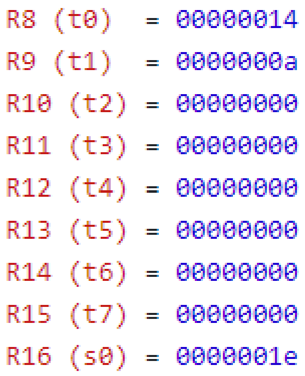
Assignment 2-6 Elaborate the actual machine code for the assembly statement 1



```
[00400024] 34080014 ori $8, $0, 20                ; 4: ori $t0, $0, 20
[00400028] 3409000a ori $9, $0, 10                ; 5: ori $t1, $0, 10
[0040002c] 01098020 add $16, $8, $9               ; 6: add $s0, $t0, $t1
```



```
R8  (t0)  = 00000014
R9  (t1)  = 0000000a
R10 (t2) = 00000000
R11 (t3) = 00000000
R12 (t4) = 00000000
R13 (t5) = 00000000
R14 (t6) = 00000000
R15 (t7) = 00000000
R16 (s0) = 0000001e
```

ori $t0, $0, 20 (note: register $8 = R8 = T0)
Memory address: 00400024
Machine code : 34080014

*Notes: **ORI** compares a **register value and a constant/immediate value**, therefore only using **one register(register source) for comparison**, and saves the result into a register target. Compared to **OR**, which **compares two register values** and saves the result into the register target.*

Goal of the Instruction: Use OR bitwise comparison between $0 (zero) and (constant/immediate) 20. Save result into register $t0 ($8). Since we're comparing it with zero, nothing is changed in the constant.
0000 0000 0001 0100 (20) |OR| 0000 0000 0000 0000 (0) = 0000 0000 0001 0100 (20)

Note: This allows us to directly save a constant/immediate value into a register with no change and with only 1 register used (this case $0).

ori $t1, $0, 10
Memory address: 00400028
Machine code : 3409000a
Goal of the Instruction: Use OR bitwise comparison between $0 (zero) and (constant/immediate) 10. Save result into register $t1 ($9). Since we're comparing it with zero, nothing is changed in the constant.
0000 0000 0000 1010 (10) |OR| 0000 0000 0000 0000 (0) = 0000 0000 0000 1010 (10)

add $s0, $t0, $t1

Memory address: 0040002c
Machine code : 01098220
Goal of the Instruction: Use addition (ADD) between registers $t0 (20) and $t1 (10). Save result into register $s0 ($16).
0000 0000 0001 0100 (20) + 0000 0000 0000 1010 (10) = 0000 0000 0001 1110 (30)

Elaboration on the Machine Code Statements

ori $t0, $0, 20
Machine code : 34080014
Operation code: “34” ORI operand = 001101 (MIPS I-Type designated Op code representation)
Register target: “08” Register Target ($8=$t0) = 1000 (Binary for 8)
Register source: “00” Register Source ($0 = zero) = 0000 (Binary for 0)
Immediate: “14” Immediate (20) = 1 0100 (Stored as Binary for 20) = 0014 (Machine code uses Hexadecimal for 20)

ori $t1, $0, 10
Machine code : 3409000a
OP code: “34” ORI operand = 001101 (MIPS I-Type designated Op code representation)
Register target: “09” Register Target ($9=$t1) = 1001 (Binary for 9)
Register source: “00” Register Source ($0 = zero) = 0000 (Binary for 0)
Immediate: “0a” Immediate (10) = 1010 (Stored as Binary for 10) = 000a (Machine code uses Hexadecimal for 10)

ori $t1, $0, 10
Machine code : 01098220
Source1: “01” 1st Register Source ($8=$t0) = 01000 (Binary for 8)
Source2: “09” 2nd Register Source ($9 = $t1) = 01001 (Binary for 9)
Destination: “82” Register Destination ($s0) = 10000 (Stored as Binary for 16) = 0082 (Machine code uses Hexadecimal for 16)
Function Code: “20” ADD = 10 0000 (Binary for ADD) = (20 = Hex)