605.611 - Foundations of Computer Architecture

Assignment 06 - MIPS Machine Code Translation

Sabbir Ahmed

March 6, 2021

1. Translate the following MIPS assembly language instructions into machine code.

   (a) `ori $t0, $zero, 15`

$$\begin{aligned}
\text{opcode(ori)} &= 0D_{16} \\
&= 001101_2 \\
\text{rs(\$zero)} &= 00000_2 \\
\text{rt(\$t0)} &= 01000_2 \\
\text{imm(15)} &\Rightarrow 000F_{16} \\
&\Rightarrow 001101\ 00000\ 01000_2 \mid 000F_{16} \\
&\Rightarrow 0011\ 0100\ 0000\ 1000 \mid 000F_{16} \\
&\Rightarrow 3408000F_{16}
\end{aligned}$$

   (b) `addi $t1, $zero, 3`

$$\begin{aligned}
\text{opcode(addi)} &= 08_{16} \\
&= 001000_2 \\
\text{rs(\$zero)} &= 00000_2 \\
\text{rt(\$t1)} &= 01001_2 \\
\text{imm(3)} &\Rightarrow 0003_{16} \\
&\Rightarrow 001000\ 00000\ 01001_2 \mid 0003_{16} \\
&\Rightarrow 0010\ 0000\ 0000\ 1001_2 \mid 0003_{16} \\
&\Rightarrow 20090003_{16}
\end{aligned}$$

(c) `add $t1, $zero $t1`

$$\begin{aligned}
\text{opcode(add)} &= 000000_2 \\
\text{rs(\$zero)} &= 00000_2 \\
\text{rt(\$t1)} &= 01001_2 \\
\text{rd(\$t1)} &= 01001_2 \\
\text{shamt} &= 00000_2 \\
\text{funct(20)} &= 20_{16} \\
&= 100000 \\
&\Rightarrow 000000\ 00000\ 01001\ 01001\ 00000\ 100000 \\
&\Rightarrow 0000\ 0000\ 0000\ 1001\ 0100\ 1000\ 0010\ 0000 \\
&\Rightarrow 00094820_{16}
\end{aligned}$$

(d) `sub $t2, $t0, $s0`

$$\begin{aligned}
\text{opcode(sub)} &= 000000_2 \\
\text{rs(\$t0)} &= 01000_2 \\
\text{rt(\$s0)} &= 10000_2 \\
\text{rd(\$t2)} &= 01010_2 \\
\text{shamt} &= 00000_2 \\
\text{funct(22)} &= 22_{16} \\
&= 100010 \\
&\Rightarrow 000000\ 01000\ 10000\ 01010\ 00000\ 100010 \\
&\Rightarrow 0000\ 0001\ 0001\ 0000\ 0101\ 0000\ 0010\ 0010 \\
&\Rightarrow 01105022_{16}
\end{aligned}$$

(e) `slt $s0, $t1, $t3`

$$\begin{aligned}
\text{opcode(slt)} &= 000000_2 \\
\text{rs(\$t1)} &= 01001_2 \\
\text{rt(\$t3)} &= 01011_2 \\
\text{rd(\$s0)} &= 10000_2 \\
\text{shamt} &= 00000_2 \\
\text{funct(2A)} &= 2A_{16} \\
&= 101010 \\
&\Rightarrow 000000\ 01001\ 01011\ 10000\ 00000\ 101010 \\
&\Rightarrow 0000\ 0001\ 0010\ 1011\ 1000\ 0000\ 0010\ 1010 \\
&\Rightarrow 012B802A_{16}
\end{aligned}$$

(f) `sra $t2, $s2, 2`

$$\begin{aligned}
\text{opcode(sra)} &= 000000_2 \\
\text{rs(0)} &= 00000_2 \\
\text{rt(\$s2)} &= 10010_2 \\
\text{rd(\$t2)} &= 01010_2 \\
\text{shamt(2)} &= 00010_2 \\
\text{funct(03)} &= 03_{16} \\
&= 000011 \\
&\Rightarrow 000000\ 00000\ 10010\ 01010\ 00010\ 000011 \\
&\Rightarrow 0000\ 0000\ 0001\ 0010\ 0101\ 0000\ 1000\ 0011 \\
&\Rightarrow 00125083_{16}
\end{aligned}$$

(g) `andi $v0, $zero, 1`

$$\begin{aligned}
\text{opcode(andi)} &= 0C_{16} \\
&= 001100_2 \\
\text{rs(\$zero)} &= 00000_2 \\
\text{rt(\$v0)} &= 00010_2 \\
\text{imm(1)} &= 0001_{16} \\
&\Rightarrow 001100\ 00000\ 00010_2 \mid 0001_{16} \\
&\Rightarrow 0011\ 0000\ 0000\ 0010_2 \mid 0001_{16} \\
&\Rightarrow 30020001_{16}
\end{aligned}$$

(h) `lw $t0, 4($t1)`

$$\begin{aligned}
\text{opcode(lw)} &= 23_{16} \\
&= 100011_2 \\
\text{rs(\$t1)} &= 01001_2 \\
\text{rt(\$t0)} &= 01000_2 \\
\text{sign_ext_imm(4)} &= 0004_{16} \\
&\Rightarrow 100011\ 01001\ 01000_2 \mid 0004_{16} \\
&\Rightarrow 1000\ 1101\ 0010\ 1000_2 \mid 0004_{16} \\
&\Rightarrow 8D280004_{16}
\end{aligned}$$

(i) `sw t2, 4($s1)`

$$\begin{aligned}
\text{opcode(lw)} &= 2B_{16} \\
&= 101011_2 \\
\text{rs(\$s1)} &= 10001_2 \\
\text{rt(\$t2)} &= 01010_2 \\
\text{sign_ext_imm(4)} &= 0004_{16} \\
&\Rightarrow 101011\ 10001\ 01010_2 \mid 0004_{16} \\
&\Rightarrow 1010\ 1110\ 0010\ 1010_2 \mid 0004_{16} \\
&\Rightarrow AE2A0004_{16}
\end{aligned}$$

2. Translate the following MIPS machine code into MIPS assembly language.

(a) 0x2010000A

$$\begin{aligned}
2010000A_{16} &= 0010\ 0000\ 0001\ 0000\ 0000\ 0000\ 0000\ 1010_2 \\
&= 001000\ 00000\ 10000_2 \mid 000A_{16} \\
001000_2 &= 08_{16} \\
&= \text{opcode(addi)} \\
00000_2 &= \text{rs(\$zero)} \\
10000_2 &= \text{rt(\$s0)} \\
\text{imm}(000A_{16}) &= 10 \\
&= \texttt{addi \$s0, \$zero, 10}
\end{aligned}$$

(b) 0x34110005

$$
\begin{aligned}
34110005_{16} &= 0011\ 0100\ 0001\ 0001\ 0000\ 0000\ 0000\ 0101_2 \\
&= 001101\ 00000\ 10001 \mid 0005_{16} \\
001101_2 &= 0D_{16} \\
&= \text{opcode(ori)} \\
00000_2 &= \text{rs(\$zero)} \\
10000_2 &= \text{rt(\$s1)} \\
\text{imm}(0005_{16}) &= 5 \\
&= \texttt{ori \$s1, \$zero, 5}
\end{aligned}
$$

(c) 0x012AC022

$$
\begin{aligned}
012AC022_{16} &= 0000\ 0001\ 0010\ 1010\ 1100\ 0000\ 0010\ 0010_2 \\
&= 000000\ 01001\ 01010\ 11000\ 00000\ 22_{16} \\
000000_2 \mid 22_{16} &= 0/22_{16} \\
&= \text{opcode(sub)} \\
01001_2 &= \text{rs(\$t1)} \\
01010_2 &= \text{rt(\$t2)} \\
11000_2 &= \text{rd(\$t8)} \\
&= \texttt{sub \$t8, \$t1, \$t2}
\end{aligned}
$$

(d) 0x00184082

$$
\begin{aligned}
00184082_{16} &= 0000\ 0000\ 0001\ 1000\ 0100\ 0000\ 1000\ 0010_2 \\
&= 000000\ 00000\ 11000\ 01000\ 00010 \mid 02_{16} \\
000000_2 \mid 02_{16} &= 0/02_{16} \\
&= \text{opcode(srl)} \\
00000_2 &= \text{rs(0)} \\
11000_2 &= \text{rt(\$t8)} \\
01000_2 &= \text{rd(\$t0)} \\
00010_2 &= \text{shamt(2)} \\
&= \texttt{srl \$t0, \$t8, 2}
\end{aligned}
$$

(e) 0x030F9024

$$
\begin{aligned}
030F9024_{16} &= 0000\ 0011\ 0000\ 1111\ 1001\ 0000\ 0010\ 0100_2 \\
&= 000000\ 11000\ 01111\ 10010\ 00000 \mid 24_{16} \\
000000_2 \mid 24_{16} &= 0/24_{16} \\
&= \text{opcode(and)} \\
11000_2 &= \text{rs(\$t8)} \\
01111_2 &= \text{rt(\$t7)} \\
10010_2 &= \text{rd(\$s2)} \\
00000_2 &= \text{shamt(0)} \\
&= \texttt{and \$s2, \$t8, \$t7}
\end{aligned}
$$