1: Add
Add $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0000
add $1, $5, $6
0000 0000 1010 0110 0000 1000 0010 0000
$d = $s + $t; advance_pc(4) => $1 = $5 + $6 => $1 = 5 + 6 = 11; $
00A60820

2: Addi
Addi $t, $s, imm
0010 00ss ssst tttt iiii iiii iiii iiii
Addi $2, $10, 2
0010 0001 0100 0010 0000 0000 0000 0010
$t = $s + imm; advance_pc(4) => $2 = $10 + 2 => $2 = 10 + 2 = 12; $
21420002

3: Addiu
Addiu $t, $s, imm
0010 01ss ssst tttt iiii iiii iiii iiii
Addiu $3, $12, 1
0010 0001 1000 0011 0000 0000 0000 0001
$t = $s + imm; advance_pc(4) => $3 = $12 + 1 => $3 = 12 + 1 = 13; $
21830001

4: Addu
Addu $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0001
addu $4, $4, $10
0000 0000 1000 1010 0010 0000 0010 0000
$d = $s + $t; advance_pc(4) => $4 = $4 + $10 => $4 = 4 + 10 = 14; $
008A2020

5: Sub
Sub $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0010
Sub $5, $20, $5
0000 0010 1000 0101 0010 1000 0010 0010
$d = $s - $t; advance_pc(4) => $5 = $20 - $5 => $5 = 20 - 5 = 15 $

02852822

6: Subu
Subu $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0011
Subu $6, $24, $8
0000 0011 0000 1000 0011 0000 0010 0011
$d = $s - $t; advance_pc(4) => $6 = $24 - $8 => $6 = 24 - 8 = 16 $
03083023

7: And
And $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0100
And $7, $23, $7
0000 0010 1110 0111 0011 1000 0010 0100
$d = $s & $t; advance_pc(4) => $7 = $23 & $7 => $7 = 23 & 7 = 7 $
02E73824

8: Or
Or $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0101
Or $8, $16, $2
0000 0010 0000 0010 0100 0000 0010 0101
0000 0010 0000 0010 0100 0000 0010 0101
$d = $s | $t; advance_pc(4) => $8 = $16 | $2 => $8 = 16 | 2 = 28 $
02024025

9: Xor //*
Xor $d, $s, $t
0000 00ss ssst tttt dddd d--- --10 0110
Xor $9, $17, $24
0000 0010 0011 1000 0100 1--- --10 0110
$d = $s ^ $t; advance_pc(4) => $9 = $17 ^ $24 => $9 = 17^ 24 = 9 $
02384826

10: Xnor
Xnor $d, $s, $t
0000 00ss ssst tttt dddd d--- --00 1100
Xnor $10, $24, $19
0000 0011 0001 0011 0101 0--- --00 1100
$d = $s ~^ $t; advance_pc(4) => $10 = $24 ^ $19 => $10 = 24 ~^ 19= -12 $
0313500C

11: Andi
Andi $t, $s, imm
0011 00ss ssst tttt iiii iiii iiii iiii
Andi $11, $21, 31
0011 0010 1010 1011 0000 0000 0001 1111
$t = $s & imm; advance_pc(4) => $11 = $21 & 31 => $11 = 21 & 31 = 21; $
32AB001F

12: Ori
Ori $t, $s, imm
0011 01ss ssst tttt iiii iiii iiii iiii
Ori $12, $20, 2
0011 0110 1000 1100 0000 0000 0000 0010
$t = $s | imm; advance_pc(4) => $12 = $20 | 2=> $12 = 20 | 2= 22;$
368C0002

13: Xori
Xori $t, $s, imm
0011 10ss ssst tttt iiii iiii iiii iiii
Xori $13, $19, 4
0011 1010 0110 1101 0000 0000 0000 0100
$t = $s ^ imm; advance_pc(4) => $13 = $19 ^ 4=> $13 = 19 ^ 4= 23; $
3A6D0004

14: SLT
SLT $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 1010
SLT $14, $0, $20
0000 0000 0001 0100 0111 0000 0010 1010
If $s < $t, $d =1; advance_pc(4) =>else $d = 0; advance_pc(4) => $14 = 1; $

0014702A

15: SLTU
SLTU $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 1011
SLTU $15, $0, $20
0000 0000 0001 0100 0111 1000 0010 1011
If $s < $t, $d =1; advance_pc(4) =>else $d = 0; advance_pc(4) => $15 = 1;
0014782B

16: SLTI
SLTI $t, $s, imm
0010 10ss ssst tttt iiii iiii iiii iiii
SLTI $16, $16, 32
0010 1010 0001 0000 0000 0000 0010 0000
If $s < imm, $d =1; advance_pc(4) =>else $d = 0; advance_pc(4) => $16 = 1;
2A100020

17: STLIU
SLTI $t, $s, imm
0010 11ss ssst tttt iiii iiii iiii iiii
SLTI $17, $17, 32
0010 1010 0011 0001 0000 0000 0010 0000
If $s < imm, $d =1; advance_pc(4) =>else $d = 0; advance_pc(4) => $17 = 1;
2A310020

18: LW //*
LW $t, offset($s)
1000 11ss ssst tttt iiii iiii iiii iiii
LW $18, 8($20)
1000 1110 1001 0010 0000 0000 0000 1000
1000 1110 1011 0010 0000 0000 0000 1000
8EB20008
$t = MEM[$s + offset]; advance_pc(4) => $18 = MEM[28] = 28 $
8E920008

19: SW //*

SW $t, offset($s)
1010 11ss ssst tttt iiii iiii iiii iiii
SW $19, 9($20)
1010 1110 1001 0011 0000 0000 0000 1001
MEM[$s + offset] = $t; advance_pc(4) => MEM[29] = $7 = 19 (because of word aligned) $
AE930009

20: LUI
LUI $t, imm
0011 11-- ---t tttt iiii iiii iiii iiii
LUI $20, 1
0011 11-- ---1 0100 0000 0000 0000 0001
$t = (imm<<16); advance_pc (4) => (1<<16) = $20 = 65536 $
3C140001

21: J //*
J target
0000 10ii iiii iiii iiii iiii iiii iiii
J 21
0000 1000 0000 0000 0000 0000 0001 0110
PC = nPC; nPC = (PC & 0xf0000000) | (target <<2) = advance_pc(8) $
08000016

22: Add //should be skipped
Add $d, $s, $t
0000 00ss ssst tttt dddd d000 0010 0000
add $1, $5, $6
0000 0000 1010 0110 0000 1000 0010 0000
$d = $s + $t; advance_pc(4) => $1 = $5 + $6 => $1 = 15 + 16 = 31;
00A60820

23: Addi
Addi $t, $s, imm
0010 00ss ssst tttt iiii iiii iiii iiii
Addi $23, $16, 2
0010 0010 0001 0111 0000 0000 0000 0010
$t = $s + imm; advance_pc(4) => $23 = $16 + 2 => $23= 1 + 2 = 3; $
22170002