

1. <Verilog>
2. <Verilog>
3. The program is compiled into an assembly language program, which is then assembled into a machine language program.
4. .
 - a. $1280 * 1024 \text{ pixels} = 1,310,720 \text{ pixels} \Rightarrow 1,310,720 * 3 = 3,932,160 \text{ bytes/frame.}$
 - b. $3,932,160 \text{ bytes} * (8 \text{ bits/byte}) / 100 \times 10^6 \text{ bits/second} = 0.31 \text{ seconds}$

5. `addi f, h, -5`
`add f, f, g`

6. `f = g + h + I;`

7. `sub $t0, $s3, $s4`
`add $t0, $s6, $t0`
`lw $t1, 16($t0)`
`sw $t1, 32($s7)`

8. r-type,
`add $s0, $s0, $s0`

9. i-type, 0xAD490020

10. r-type
`sub $v1, $v1, $v0`
 0x00621822

11. i-type
`lw $v0, 4($at)`
 0x8C220004

12. opcode would be 8 bits, rs, rt, rd fields would be 7 bits each

opcode would be 8 bits, rs and rt fields would be 7 bits each

more registers → more bits per instruction → could increase code size
 more registers → less register spills → less instructions
 more instructions → more appropriate instruction → decrease code size
 more instructions → larger opcodes → larger code size

13. 0xBABEFEF8
0xAAAAAAAA0
0x00005545

14. I-type

```
addi $t2, $t2, -1  
beq $t2, $0, loop
```

15. 20

```
i = 10;  
while( i > 0 ) {  
    B += 2;  
    --I;  
}
```

5N + 2 instructions