汇编语言例题

寄存器

1. 判断：
   1. In order to use the saved registers (s0-s11) in a function, we must store their values before using them and restore their values before returning.正确，见讲义
   2. 错误，jalr 和 jal不同，会储存返回地址
   3. 对的，但是通常有保护所以我们常识上不会访问
   4. 错误，详情请见64位附加指令

机器码汇编语言转换

1. 0xE2952023对应的指令是什么 1110001_01001_01010_010_00000_0100011 立即数 111000100000=-480 `sw x9,-480(x10)`
2.

Address(Hex)	inst	machine code
200010	Loop: add x18,x19,x20	0x01498933
200014	jal x0,L1	0x0540006f
...		
20001c	bne x20,x29,Loop	0xffde1ae3
.....		
200070	L1	

   1. add : func7 = 0000000 rs2=10100 rs1=10011 func3=000 rd=10010 op = 0110011
   2. jal x0,0x5c imm[20]=0 imm[10:1]=0000101010 imm[11]=0 imm[19:12]=00000000 rd=00000 opcode=1101111
   3. bne x28,x29,-12 imm[12]=1 imm[10:5]=111111 rs2=11101 rs1=11100 func3=001 imm[4:1]=1010 imm[11]=1 opcode=1100011

汇编功能

1. `sltu true,x11,x20`
2. `beq`+`jal`

```
   lui t0, 0x12345    # 高20位 0x12345，低12位在lui中为0x000
   ori t0, t0, 0x800 # 将低12位设为0x800
   lw ,0(t0) # 如果你准备在这一步直接用加法请务必注意负数问题
   ```

4. `xori rd,rs, -1`
5. `sltu rd,x0,rs`

C语言转汇编

1. 此为64位系统

```
fib:
  beq     x10,     x0, done   // If n==0, return n
  addi     x5,     x0, 1
  beq     x10,     x5, done    // If n==1, return n
  addi     x2,     x2, -16   //    Allocate 2 words of stack space
  sd     x1,     0(x2)   // Save the return address
  sd     x10,     8(x2)   // Save the current n
  addi     x10,     x10, -1     // x10 = n-1
  jal     x1,     fib   // fib(n-1)
  ld     x5,     8(x2)   // Load old n from the stack
  sd     x10,     8(x2)   // Push fib(n-1) onto the stack
  addi     x10,     x5, -2   // x10 = n-2
  jal     x1,     fib   // Call fib(n-2)
  ld     x5,     8(x2)   // x5 = fib(n-1)
  add     x10,     x10, x5   // x10 = fib(n-1)+fib(n-2)
  // Clean up:
  ld     x1,     0(x2)   // Load saved return address
  addi     x2,     x2, 16   // Pop two words from the stack
done:
  jalr     x0,     x1
```