UFPR – Bacharelado em Ciência da Computação
CI210 – Projetos Digitais e Microprocessadores
Trabalho Semestral — 2019-2 14 de novembro de 2019

O Mico XII é um processador extremamente simples que, a cada ciclo, busca e executa uma instrução de lógica/aritmética, uma instrução de controle de fluxo (salto/desvio), ou uma instrução de acesso à memória. Sua tarefa é preencher a memória do circuito de controle da versão XII do Mico. O conjunto de instruções do Mico XII está definido na Tabela 1.

Tabela 1: Instruções do Mico XII.

opcode	instrução	semântica	comentário
0	`nop`	`--`	*no operation*
1,2,3	`op c, a, b`	`R(c) ← R(a) op R(b)`	`op ∈ [add, sub, mul]`
4,5,6	`op c, a, b`	`R(c) ← R(a) op R(b)`	`op ∈ [and, or, xor]`
7	`not c, a`	`R(c) ← not(R(a))`	complemento
8	`addi c, a, K`	`R(c) ← R(a) + extSinal(K)`	+ constante aritmética
9	`sra c, a, b`	`R(c) ← R(a) >> R(b)`	desloc aritm direita
a	`ld c, K(a)`	`R(c) ← M[ R(a) + extSinal(K) ]`	*load from memory*
b	`st b, K(a)`	`M[ R(a) + extSinal(K) ] ← R(b)`	*store to memory*
c	`bran a, b, E`	`IP ← ((R(a)==R(b))? E : IP+1)`	desvio condicional
d	`jal E`	`IP ← E : r15 ← IP+1`	*jump and link*
e	`jr a`	`IP ← R(a)`	*jump register*
f	`halt`	`IP ←IP + 0`	termina a simulação

O diagrama de blocos do Mico XII é mostrado na Figura 1. O circuito de dados consiste de uma Unidade de Lógica e Aritmética (ULA), de um Bloco de Registradores (R), e circuitos auxiliares. O circuito de controle consiste de um apontador de instrução (registrador chamado de *instruction pointer*, IP), um somador, de uma memória de instruções (MI), de um comparador de igualdade (que não é mostrado no diagrama) e de uma memória de dados (M).

No diagrama, e na Tabela 1, `a,b,c` são nomes de registradores (endereços dos registradores) enquanto que `A,B,C` são os conteúdos dos respectivos registradores (`R(a)=A`). O registrador 0 contém a constante zero (`R(0)=0`); escritas neste registrador não tem nenhum efeito.

A ULA executa as operações de lógica e aritmética definidas para as instruções. O bloco de registradores contém 16 registradores de 32 bits, duas portas de leitura (A e B) e uma porta de escrita (C). Este bloco de registradores é similar ao da Seção 7.7.2 das notas de aula, ou da Seção 2.9.1 do livro verde.

A memória M se comporta como um circuito combinacional nas leituras – a porta ρ sempre mostra o conteúdo da posição apontada pelo endereço – e como um registrador nas escritas – a posição de memória apontada pelo endereço é atualizada com o valor na porta ω na borda de subida do relógio, se a escrita estiver habilitada. Ao contrário da memória do MIPS, a memória do Mico é endereçada palavra a palavra.

Você recebeu um *testbench* com a infraestrutura para testar seu projeto. O arquivo `mico.vhd` contém o código VHDL com o modelo do Mico.

A tabela de controle (`ctrl_table`) deve ser alterada de acordo com a funcionalidade de cada instrução. Todos os sinais de controle são ativos em `'1'`. Sua tarefa é compreender o modelo e então preencher a tabela de controle para implementar a funcionalidade de cada instrução.

Você deve traduzir o programa em C da Figura 2 para *assembly* do Mico XII. O resultado computado deve ser exibido no *display*. Um valor escrito com um (`st`) no endereço `0xffff` é exibido no *display*. Note que o apontador de pilha deve ser inicializado num endereço *distinto* de `0xffff`.

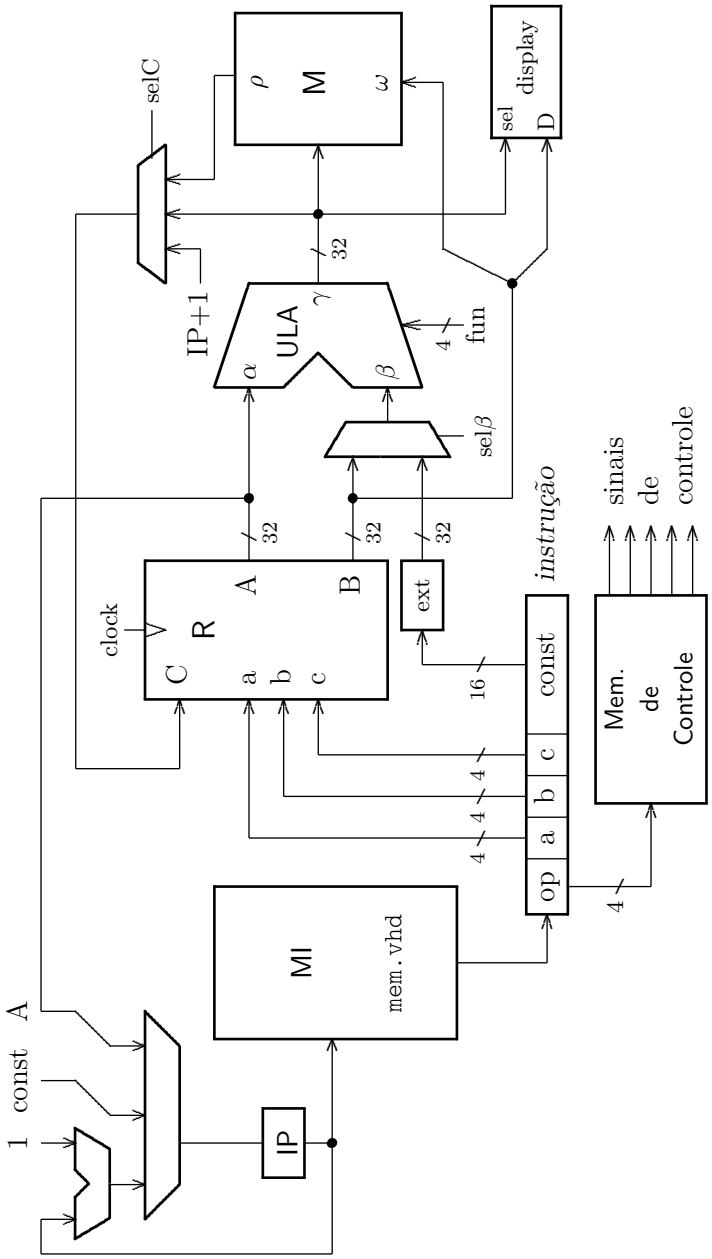

Figura 1: Circuito de dados e de controle do Mico XII.

Para tanto você deve (i) traduzir o programa para o assembly do Mico XII; e (ii) traduzir o assembly para binário; e (iii) editar o arquivo `mem.vhd`. Para traduzir o programa C para *assembly* do Mico, use uma convenção similar à do MIPS, indicado na Tabela 2.

O conteúdo de `mem.vhd` que vocês receberam contém um programa que testa umas poucas instruções – complete-o para testar todas as 16 instruções.

Você deve sobrescrever este programa com sua versão do binário do programa de testes.

Da execução:

1. Assegure-se de que entendeu a especificação antes de iniciar a programação da memória de controle e em *assembly*;
2. o trabalho pode ser efetuado em duplas;
3. copie o arquivo `www.inf.ufpr.br/roberto/ci210/trab2019-2.tgz` para sua área de tra-

```
int log2(int n) {
   if (n < 2) then
      return 0;
   else
      // divisão com deslocamento
      return (1 + log2(n/2));
}

// empilhe o que deve ser empilhado
int power(int n,int exp) {
   if (exp > 1)
      return (n * power(n,exp-1));
   else
      return (n);
}
```

```
void main(void) {
   int i;

   for(i=4; i > 0; i=i-1) {
     // store to 0xffff
     display(i);
     display(log2(power(i, 4)));
   }
   // instrução halt
   halt();
}
```

Figura 2: Programa de teste.

Tabela 2: Convenção para uso de registradores.

reg.	nome	função
0	r0	sempre zero
1	v0	valor de retorno de função
2	a0	primeiro argumento
3	a1	segundo argumento
4..13	r4..r13	registradores de uso geral
14	sp	apontador de pilha
15	ra	endereço de retorno

balho. O arquivo `LEIA-ME` contém descrição dos conteúdos de `trab2019-2.tgz` . Estude o *script* `run.sh` e código VHDL antes de usá-los;

4. PLÁGIO NÃO SERÁ TOLERADO. É do interesse de todos que alunos conversem sobre o projeto mas cada grupo deve escrever seu próprio código.

Dos produtos:

1. Relatório em papel A4, com letras em 11 pontos, espaço simples, formatação simples, contendo os nomes dos componentes do grupo e o conteúdo do arquivo `mem.vhd`. No interesse da proteção da vida, esse relatório não deve passar de duas páginas (frente e verso). Se o relatório ultrapassar duas páginas, use letras de até 9 pontos;
2. envie por e-mail para seu professor os arquivos `mem.vhd` e `mico.vhd`, em mensagem com cabeçalho "mico 12";
3. todos os programas serão compilados antes de serem simulados na avaliação.

Referências

[RH12] *Sistemas Digitais e Microprocessadores*, R.A.Hexsel, 2012, Editora da UFPR.

Histórico das Revisões:
`14nov2019`: primeira versão da especificação; versão preliminar do código VHDL.