Universidade Católica Dom Bosco

Curso de Bacharelado em Engenharia de Computação

Disciplina Tópicos Avançados em Sistemas de Computação

Relatório de Implementação do Microprocessador ARM7 e Otimizações de Compilador

Academico: Horácio Ortiz

Prof. Dr. Ricardo Santos

UCDB - Campo Grande - MS - Nov/2007

Capítulo 1

Desenvolvimento da Aplicação

1.1 O que foi implementado.

A iplementação iniciou-se pelo estudo dos documentos que se encontram na página da disciplina. Em seguida começou o desenvolvimento dos arquivos que dão origem ao simulador do microprocessador ARM7. O primeiro arquivo desenvolvido foi o arm7.ac,conforme mostra a figura 1, que contém a descrição do ARM7 incluindo o tamanho da memória, do tamanho da palavra da arquitetura e a descrição dos registradores. O segundo arquivo foi o arm7-isa.ac,conforme mostra a figura 2, onde esta descrito o formato das instruções e sintaxe. Por ultimo o arquivo arm7-isa.cpp, que contém a descrição do comportamento (behavior) das instruções. A figura 3 mostra o comportamento da instruão ADD para exemplificar um comportamento.

```
AC_ARCH(arm7){

     ac_wordsize 32;
     ac_mem   DM:32M;
     ac_regbank RB:32;

     ac_format F_IF_ID ="%npc:32";
     ac_format F_ID_EX ="%npc:32 %data1:32 %data2:32 %data3:32 %imm:8 %rn:4 %rd:4 %rs:4 %rm:4
rotate:4";
     ac_format RBStatus= "%n:1 %z:1 %c:1 %v:1 %q:1 %DMN:18 %I:1 %F:1 %T:1 %M4:1 %M3:1 %M2:1 %M
M0:1";

     ac_reg<F_IF_ID>  IF_ID;
     ac_reg<F_ID_EX>  ID_EX;
     ac_reg<RBStatus> CPSR, SPSR;

     ac_stage   IF,ID,EX;

ARCH_CTOR(arm7) {

    ac_isa("arm7_isa.ac");
    set_endian("big");
```

Figura 1.1: Figura 1 - Corpo do arquivo arm7.ac.

As instruções implementadas foram as dos tipos Data Processing instructions (DP), Branch instructions(B), Multiplications instructions(M) e load/store instructions(LS). Essas instruções foram:

```
AC_ISA(arm7){

  ac_format Type_DP = "%cond:4 %func1:2 %I:1 %opcode:4 %s:1 %rn:4 ...

  ac_format Type_Multiply  = "%cond:4 %func2:7 %s:1 %rn:4 %rd:4 ...

  ac_instr<Type_DP> sys_call, add, adc, sbc, sub, rsb, rsc, and1...

  ISA_CTOR(arm7)

        add.set_asm("add %rd, %rn, %operand2");
        add.set_decoder(opcode=0x04, func1=0x00);

        adc.set_asm("adc %rd, %rn, %operand2");
        adc.set_decoder(opcode=0x05, func1=0x00);


        mov.set_asm("mov %rd, %operand2");
        mov.set_decoder(opcode=0x0d, func1=0x00);

        mul.set_asm("mul %rd, %rm, %rs");
        mul.set_decoder(func2=0x00);
```

Figura 1.2: Figura 2 - Como sao definidas alguns tipos de instruções.

```
void ac_behavior( add ){

  switch( stage ) {
  case IF:
            break;

  case ID:
    break;

  case EX:  result=ID_EX.data1 + ID_EX.data2;
            RB.write(rd, result);

            if(s==1){
                  if(rd==PC){
                        copySPSR();
                  }
            }
            else if(s==1){
                  CPSR.n= RB.read(rd) >> 31;

                  if(RB.read(rd)==0)CPSR.z=1;
                  else CPSR.z=1;

                  CPSR.c=((ID_EX.data1 & ID_EX.data2 & 0x80000000)|(~result & (ID_EX.data1 | ID_EX.data2)) & 0x8000000

                  CPSR.v=((ID_EX.data1 & ID_EX.data2 & ~result & 0x80000000)|(~ID_EX.data1 & ~ID_EX.data2 & result &
0x80000000) );
            }

     break;

  default:
    break;
  }
};
```

Figura 1.3: Comportamento de um ADD

```
Data Processing:"%cond:4 %func1:2 %I:1 %opcode:4 %s:1 %rn:4 %rd:4

 [ %rotate_imm:4 %immed_8:8 | %code:8 %rm:4 | %shift_imm:5 %code1:3

  %rm1:4 | %rs:4 %code2:4 %rm2:4 ]";
```

Instruções. add, adc, sbc, sub, rsb, rsc, and1, orr, eor, bic, tst, teq, cmp, cmn,mov,mvn .

Multiplications instructions:

```
 %cond:4 %func2:7 %s:1 %rn:4 %rd:4 %rs:4 %constante:4 %rm:4";
```

Instruções. mul

1.2 O que falta implementar.

Para a geração de um simulador completo do microprocessador ARM7 falta implementar varias outras instruções de todos os tipos de intruções que são descristas no data sheet do microprocessador.

Capítulo 2

A Ferramenta

A ferramenta gerada foi um protótipo de um simulador do microprocessador ARM7 que executa algumas instruções desta arquitetura ARM, usada principalmente em sistemas embarcados.

2.1 A Sintaxe para execução

Para a execução da ferramenta implementada, são necessárias alguns passos e requisitos de sistema. Os requisitos de sistema são:

1. Ter instalado a ferramenta ArchC;

2. ter a biblioteca SystemC;

3. e um arquivo de teste no formato hexadecimal(teste.h).

Os passos para executar o arquivo teste é:

1. Localizar o diretorio do accsim;

2. entao colocar o comando accsim arm7.ac -dy –load=./teste.h;

3. depois dar o comando make -f Makefile.archc;

4. verificar se o arquivo arm7.x foi gerado

5. por fim executar ./arm7.x

Capítulo 3

Conclusão

3.1 Bugs/Falhas detectados e não corrigidos.

Durante a construção do simulador foram encontrados vários erros e/ou falhas, alguns corrigidos outros não. Alguns erros como o de dependencia de pacotes na máquina de desenvolvimento puderam ser facilmente corrigidos. Outros erros que dizem respeito a compilação dos arquivos gerados até a geração do arquivo final executável foram sendo corrigidos durante a implementação.

O principal erro não corrigido foi o da saída final, que não emite o resultado das operações contidas do arquivo teste.h. Esse erro invalida toda o simulador, pois deixa claro que existe erros no códigos dos arquivos *.ac e .cpp.

3.2 Dificuldades encontradas durante o desenvolvimento do trabalho.

A primeira e principal grande dificuldade foi a falta de tempo. Durante o período de desenvolvimento também estava em desenvolvimento paralelamente o projeto de graduação e a divisão de tempo atrapalhou um pouco. Outra dificuldade também foi o material encontrado para leitura e orientação estar todo em inglês.

Durante a parte de escrita dos códigos dos arquivos do simulador, foram econtradas dificuldades como a de manipular variaveis já prescristas do ambiente. Passar os comportamentos de cada instrução implementada, do data sheet para o código, também foi complicado. Setar as flags em determinada operação e diferenciar as varias flags.

Por fim a transformação das diferentes instruções que formaram o conjunto de teste do simulador foram uma das grandes dificuldades deste trabalho.

3.3 Conclusões Finais

O desenvolvimeto deste simulador utilizando a ferramenta ArchC depende, exclusivamente, da complexidade da arquitetura e do conjunto de instrução desejavél para a execução. De certo modo a contrução do simulador ARM7 foi custoso e levou cerca de dois meses para ser construido.

O estudo de aplicações em que o ARM7 pode ser utilizado e o quanto pode ser interessante dar continuidade aos estudos desta arquitetura deixa claro através deste trabalho que ainda existe muito a ser pesquisado e desenvolvido.