

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia de Sistemas Eletrônicos - PSI

PSI-2553- Projeto de Sistemas Integrados

O Processador Plasma - Parte 1

Objetivos

Esta experiência visa familiarizar o estudante com o código *assembly* e o uso de algumas ferramentas de CAD relacionadas ao processador MLITE que faz parte do sistema Plasma. Este processador segue a mesma arquitetura do MIPS introduzida nas aulas de teoria.

Serão realizadas as tarefas de compilação de código C, o qual gera o respectivo código *assembly*, seguida da simulação do código utilizando o simulador do conjunto de instruções (ISS) associado ao MLITE. A aplicação a ser testada será o algoritmo que realiza o cálculo do número de Fibonacci.

Compilação e Montagem

O **compilador** converte um programa em linguagem de alto nível C para a linguagem *assembly* (arquivos F.lst e F.map na figura 1) e o **montador** realiza a tarefa de conversão do *assembly* para o código binário.

O compilador e o montador realizam as tarefas indicadas na Figura 1.

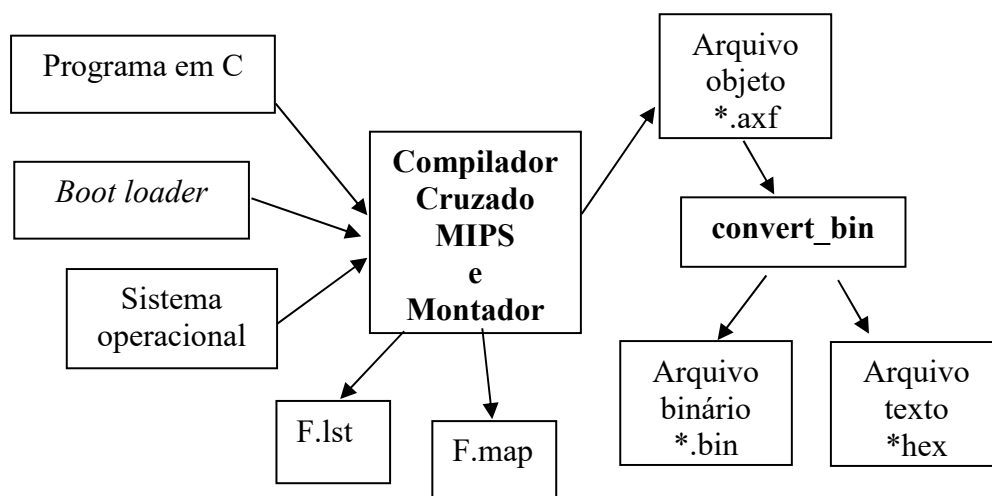


Figura 1. Fluxo de compilação e montagem a partir da linguagem C no Plasma

O programa em linguagem C é a aplicação que se deseja implementar no Plasma (Fibonacci nesta experiência). O *boot-loader* contém a inicialização dos registradores do Plasma enquanto o sistema operacional cuida dos periféricos: blocos *ethernet*, UART e *general purpose* IO (veja apostila de teoria). O arquivo binário corresponde à imagem pronta para ser carregada na memória física, enquanto o arquivo texto permite ao desenvolvedor verificar o código *assembly* assim como seu equivalente código hexadecimal.

Outra opção possível seria desenvolver o código em linguagem *assembly* e a partir dele realizar a montagem. Esta opção não será utilizada nesta experiência.

Parte prática

Preparando os arquivos de aínese, compiladores e ferramentas do Plasma

- 1) Criar a pasta **X:\psi2553\tut_mips**.
- 2) Copiar para **X:\psi2553\tut_mips** o arquivo **mips_tools.zip** que está em **Rede=> Newserverlab\psi2553\tut_mips**.
- 3) No subdiretório **X:\psi2553\tut_mips** extraia as pastas de **mips_tools.zip** (utilize o software 7zip por exemplo).

Duas pastas deverão aparecer:

X:\psi2553\tut_mips\tools

X:\psi2553\tut_mips\gccmips_elf.

Atenção: a compilação e a montagem será feita no subdiretório **X:\psi2553\tut_mips\tools**. Os programas serão usados via linhas de comando (diretamente no DOS).

Compilando o programa em C

- 1) Leia e entenda os seguintes arquivos disponibilizados em **X:\psi2553\tut_mips\tools**
 - a. **Fibonacci.c** (programa fonte). Neste arquivo altere o número de Fibonacci que deve ser calculado na chamada à função **fibonacci**. Utilize o valor: **(10+##_USP mod30)**.
 - b. **boot.asm** (código de *boot* em *assembly* que é mesclado ao programa fonte)
 - c. **no_os.c** (código em C de operações de sistema, em ausência de SO)
 - d. **makefile** (*script* para compilação programada e incremental)
- 2) No Windows, selecione **Start→All Programs→Accessories→Command Prompt**. Uma janela de terminal (prompt de comando) se abrirá.
- 3) No prompt de comando, digite **C: <enter>**.
- 4) Tecle **mkdir C:\tmp** para criar o sub-diretório **C:\tmp**. Ignore se a pasta já existir.
- 5) Tecle **X: <enter>**.
- 6) Digite o comando **cd X:\psi2553\tut_mips\tools** para acessar esta pasta
- 7) Digite **set path=.** <enter>.
- 8) Tecle **gmake Fibonacci**. A compilação e montagem, como descrita na parte teórica, será realizada. (Observação: gmake roda o arquivo makefile)

Terminada a compilação, verifique que foram criados os seguintes arquivos:

- test.map (mapeamento de endereços e funções)
 - test.lst (listagem de comandos *assembly* do algoritmo de Fibonacci + outros comandos)
 - arquivos binários.
- 9) Observe o arquivo **test.map**.

Anote no seu roteiro: os endereços de

- *main*
- função *Fibonacci*

- 10) Observe o arquivo **test.lst**. Entenda o código *assembly* e verifique que a função *Fibonacci* é executada. Você pode utilizar o arquivo *mips_ref.pdf* disponível no \\newserverlab\psi2553 para se orientar em relação as instruções.

Anote no seu roteiro: o(s) código(s) referente(s) a

- Início do Main
- Fim do Main
- Início da função Fibonacci
- Fim da função Fibonacci
- Ponto de chamada da função Fibonacci no Main
- Passagem do argumento para a função Fibonacci
- Retorno do valor calculado na função Fibonacci para o Main (onde ele é armazenado?)

Simulando através de um ISS (*Instruction Set Simulator* ≠ de simular em VHDL)

- 1) Tecle na janela DOS **gmake run** . A simulação será apresentada com algumas opções.
- 2) Inicialmente tecle a opção 1 seguidamente e verifique que o programa *assembly* é executado passo a passo. Observe como variam os valores dos registradores (apresentados nas 4 linhas iniciais). Isto faz com que o processador avance o PC para a próxima instrução. Você perceberá que haverá um momento em que a execução ficará retido em um *loop*
- 3) Para acelerar a execução do *loop* podemos utilizar a opção 3 e em seguida digite 120 <enter>.
- 4) Teclando a opção 1, prossiga com mais alguns passos e assim se observará o *loop* de inicialização sendo finalizado e o programa *main* sendo chamado.

Anote no seu roteiro: Copie o trecho da simulação quando o PC estiver

- Chamando o main.
- Passando o número de *Fibonacci* a ser calculado.
- Executando a primeira instrução do Fibonacci.
- Retornando o resultado do seu número de Fibonacci (destacando na imagem os registradores do processador envolvidos no retorno).

5) Compare as simulações com o resultado esperado

Anote no seu roteiro: explique se as simulações estão coerentes com o esperado

- 6) Para finalizar a simulação pressione 9 ou ctrl+C.
- 7) Se quiser reiniciar a simulação, após ter finalizado digite novamente **gmake run**.