Centro Federal de Educação Tecnológica de Minas Gerais
CEFET-MG

DEPARTAMENTO DE COMPUTAÇÃO (DECOM)
LABORATÓRIO DE ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES I
Professora: Juliana Santiago Teixeira

Aluno: Igor Luciano de Paula

PRÁTICA 4 - Projeto de um Processador: planejamento do conjunto de instruções

1) Explique o que o programa embarcado deverá fazer.

R.: O programa embarcado tem o propósito de realizar a criptografia e descriptografia assimétrica de dados. A criptografia assimétrica tem como característica utilizar duas chaves para a sua realização, uma para cifrar e outra para decifrar os elementos.
O modelo do método usado para realizar a criptografia deve ser tal que traga a segurança das informações e inviabilize o acesso as mesmas em caso de interceptação por terceiros.
Caso os dados sejam interceptados, sem chave para a criptografia, o invasor não será capaz de traduzir as informações a menos que execute um "Backtracking". O Backtracking é um algoritmo de tentativa e erro, comumente usado em ataques de "Brute Force" (força bruta). Este tipo de invasão tem como característica tentar realizar todas as combinações possíveis para obter a chave. Caso a chave seja grande o problema é considerado intratável (*O* (exponencial)), pois para o atacante achar a chave, será necessário um grande custo computacional, podendo demorar anos para encontrar a mesma.

2) Apresente a lista de instruções suportadas pelo seu processador.

R.: O processador terá oito instruções. Sendo elas:

- jal
- jr
- add
- bne
- sw
- lw
- sub
- slt

3) Explique a operação realizada por cada uma das instruções.

- jal: (jump and link) desvia para o endereço ou label especificado e armazena em $ra o endereço de retorno (PC + 4).

- jr: (jump register) desvia para o endereço de retorno contido em $ra.

- add: (addition) adiciona dois elementos e armazena em um terceiro.

- bne: (branch if not equal) desvia se os registradores não forem iguais.
- sw: (store word) armazena palavra.
- lw: (load word) carrega palavra.
- sub: (subtraction) realiza a subtração de dois elementos.
- slt: (set less than) seta o registrador base como 1 se o primeiro registrador for menor que o segundo.

4) Mostre a representação (sintaxe) em assembly de cada instrução.

- jal: jal funct
- jr: jr $ra.
- add: add $t0, $s0, $s1
- bne: bne $s0, $s1, funct
- sw: sw $t0, 8($s0)
- lw: lw $t0, 8($s0)
- sub: sub$t0, $s0, $s1
- slt: slt $t0, $s0, $s1

5) Indique o formato binário de cada uma das instruções, apontando o tamanho (em número de bits) e a função de cada campo das instruções.

R.: O formato para as instruções neste projeto serão: 000 0 00 00

- jal: 000 00000 - os três primeiros bits se refere a operação que será realizada e os últimos 5 o endereço para o qual será desviado o fluxo.
- jr: 001 00000 - os três primeiros bits se refere a operação que será realizada e os últimos 5 o endereço para o qual será desviado o fluxo.
- add: 010 0 01 10 - os três primeiros bits se refere a operação que será realizada. O próximo bit é utilizado para receber o resultado. Os próximos dois pares de bits serão os operandos.
- bne: 011 0 00 00 - os três primeiros bits se refere a operação que será realizada. O próximo bit é utilizado como zero para a comparação. Os próximos pare será o operando. E o ultimo par será o endereço para desvio.
- sw: 100 0 0 000 - os três primeiros bits se refere a operação que será realizada. Os próximos dois bits se refere aos operandos, onde o primeiro é o registrador base para o cálculo de endereço e o segundo o registrador que receba a a palavra. Os últimos três bits representam a constante para o cálculo de endereço.

- lw: 101 0 0 000 - os três primeiros bits se refere a operação que será realizada. Os próximos dois bits se refere aos operandos, onde o primeiro é o registrador base para o cálculo de endereço e o segundo o registrador que receba a a palavra. Os últimos três bits representam a constante para o cálculo de endereço.

- sub: 110 0 01 10 - os três primeiros bits se refere a operação que será realizada. O próximo bit é utilizado para receber o resultado. Os próximos dois pares de bits serão os operandos.

- slt: 111 0 01 10 - os três primeiros bits se refere a operação que será realizada. O próximo bit é utilizado para receber o resultado. Os próximos dois pares de bits serão os operandos.

5) Justifique todas as suas decisões de projeto.

R.: As decisões de projeto visaram utilizar o máximo do coprocessador para que com o mesmo exista a possibilidade de se implementar o algoritmo que resulte em um modelo de criptografia mais complexa possível, dificultando com que os dados sejam decifrados por um invasor.