

Computador Simplificado

Conceitos

p.ex. FLUXO de execução

Conceitos – Computador Simplificado

Memória com 16 posições: E0, E1, ...E15

Dispositivo de entrada: cartões

Dispositivo de saída: impressora

Processador/ULA

Arquivo

- Vários conceitos dos computadores não existem no Computador Simplificado, por exemplo o conceito de arquivo, apesar de serem usados de forma “implícita”. Em um computador existe um mecanismo que irá fazer a **carga** de um “programa” na “memória” do computador, em geral, a partir de um arquivo. No Computador Simplificado iremos assumir que as “instruções” e os “dados” necessários para a execução de um programa <aparecem> de forma adequada na memória.

Iniciação do Computador Simplificado

- O computador simplificado além dos itens anteriores tem também uma anotação i que identifica a posição E_i (“atual”) onde ele irá ler 0’s e 1’s para serem interpretados como uma “instrução”
- As posições da Memória do computador simplificado devem ser “escritas” com 0’s e 1’s de forma adequada (programação).

Ciclo de funcionamento

- “Ligado” o processador:
- Passo 0: O programa é colocado na memória; E0 é anotado como sendo posição atual;
- Passo 1: Pegue os 0’s e 1’s que está na posição anotada atual;
- Passo 2: Anote a posição seguinte como atual ;
- Passo 3: Decodifique os 0’s e 1’s identificando a instrução correspondente;
- Faça o que manda a instrução;
- Passo 4: Volte para o Passo 1.

ciclo

- Busca
 - Decodificação
 - Execução
-
- Fetch
 - Decode
 - Execute

Memória

- Endereço/posição e dado/valor
- Leitura: a partir de um endereço (posição) obter o dado
- Escrita: a partir de um endereço (posição) e um dado (valor)
- ROM, PROM, EPROM, EEPROM etc

Instruções

- leia cartão e guarde em E_i
- copie E_i em E_j
- some(subtraia, multiplique,...) E_i e E_j e guarde em E_k
- vá para E_i
- se E_i (maior, menor, maior ou igual,...) E_j vá para E_k .
- imprima E_i
- pare

Exemplos

- Subtraia de E15 E15 e guarde em E15
- Some E15 e E15 e guarde em E15
- Divida E15 por E15 e guarde em E15
- Multiplique E15 por E15 e guarde em E15

Algoritmo

- Um algoritmo é um procedimento que permite a um agente resolver um problema. Um algoritmo se constitui de uma seqüência de passos e um algoritmo deve terminar depois de seguidos um número finito de passos. Cada passo de um algoritmo deve ser definido de maneira precisa envolvendo ações que podem ser cumpridas de maneira rigorosa. Um algoritmo pode ter nenhuma, uma ou mais entradas, isto é valores que estão disponíveis conforme as especificações dos passos do algoritmo. Um algoritmo pode ter uma ou mais saídas, isto é valores que são produzidos através da execução do algoritmo. Cada um dos passos de um algoritmo deve poder ser executado de maneira exata e em um tempo finito.

Programa

- Uma das possíveis representações de algoritmos é um programa de computador, o agente neste caso é um computador.
- Um programa consiste em uma sequência de instruções e constantes que podem ser armazenadas na memória do computador para poder atingir um determinado fim. Um programa tem um aspecto estático correspondente à sequência de instruções e constantes e um aspecto dinâmico correspondente à execução das instruções. O aspecto dinâmico de um programa depende não só das instruções mas também dos valores que o programa obtém do mundo externo (leia cartão e guarde em Ei). Os valores lidos pelo programa são denominados **Entrada**. Os programas podem interferir no mundo exterior através de instruções especiais (imprima Ei). Instruções que produzem modificações no mundo externo ao computador e são denominadas instruções de **Saída**.
- Ser humano?

Desenvolvimento de Programas

- Os requisitos são levantados
- Uma solução é investigada e descrita
- Um programa correspondente à solução é codificado em uma linguagem de programação;
- O programa é testado e modificado até que seja considerado correto.

Uso de programas

- Sempre que quisermos usar um programa o primeiro passo é “carregar” as instruções e dados do programa. Carregar o programa consiste em copiar as instruções e dados de algum dispositivo de armazenamento tal como Disco rígido ou CD (arquivo) para a memória do computador. O processador só executa instruções que estejam na memória do computador. Se as instruções estiverem em algum outro dispositivo de armazenamento é preciso que elas sejam primeiro transferidas para a memória. No Computador Simplificado não é definido arquivo e carga. O uso do programa assume que ele <aparece> na memória.

Programa 1

- E0: leia cartão e guarde em E15
- E1: leia cartão e guarde em E14
- E2: se E15 maior que E14 vá para E5
- E3: imprima E14
- E4: vá para E6
- E5: imprima E15
- E6: pare

Fluxo de controle/execução

- O aspecto dinâmico de um programa pode ser descrito pelo fluxo de controle ou execução. A seqüência inicial do fluxo de controle do Programa 1 corresponde a:
 - E0, E1, E2 ...
- A instrução em E2 corresponde a um desvio condicional e neste ponto o fluxo depende da Entrada do programa. Vamos assumir que o programa tenha lido o valor 4 na instrução correspondente a E0 e tenha lido o valor 3 na instrução correspondente a E1. Neste caso o fluxo de E2 passa para E5, e termina em E6.
 - E0, E1, E2, E5, E6.
- Se considerarmos que os valores lidos foram 3 em E0 e 4 em E1 (são os mesmos valores considerados antes mas agora invertidos!), o fluxo passa de E2 para E3 e daí para E4 e E6. Neste caso o fluxo corresponde a
 - E0, E1, E2, E3, E4, E6.
- O Programa 1 tem apenas 2 fluxos distintos.

Fluxos de execução

- Número de fluxos
- Tamanho de fluxos
- Menor programa/Menor fluxo
- Menor programa/Maior fluxo

Programa 2

- E0: leia cartão e guarde em E13
- E1: se E13 igual a E15 vá para E4
- E2: se E13 igual a E14 vá para E4
- E3: vá para E0
- E4: pare
- E14: 0
- E15: 1

Fluxos do Programa 2

- Os fluxos de execução são da forma:
- E0, E1, E4
- E0, E1, E2, E4
- E0, E1, E2, E3, E0, E1, E4
- E0, E1, E2, E3, E0, E1, E2, E4
- E0, E1, E2, E3, E0, E1, E2, E3, E0, E1, E2,
- ...
- Infinitos fluxos!

Linguagem de Máquina

- leia Cartão e guarde em Ei
 - copie Ei em Ej:
 - vá para Ei
 - some(subtraia, multiplique,...)
Ei e Ej e guarde em Ek:
 - se Ei (maior, igual, ...) Ej vá
para Ek
 - imprima Ei
 - pare:
- 111 00000000 IIII
 - 111 0001 IIII JJJJ
 - 111 00000001 IIII
 - yyy IIII JJJJ KKKK
some yyy=000;
subtraia yyy=001;
multiplique yyy=010;
divida yyy=011;
 - yyy IIII JJJJ KKKK
maior yyy=100;
igual yyy=101;
menor yyy=110;
 - 111 00000010 IIII
 - 111 00000011 0000

Código binário

- 0,1
- 00, 01, 10, 11
- 000, 001, 010, 011, 100, 101, 110, 111
- 0=0000 1=0001 2=0010 3=0011
- 4=0100 5=0101 6=0110 7=0111
- 8=1000 9=1001 A=1010 B=1011
- C=1100 D=1101 E=1110 F=1111

Programa 3

- E0:leia cartão e guarde em E15;
- E1:leia cartão e guarde em E14
- E2:se E15 igual a E12 vá para E6
- E3:some E14 e E13 e guarde em E13
- E4:some E11 e E12 e guarde em E12
- E5:vá para E2
- E6:imprima E13
- E7:Pare
- E11:1; E12:0; E13:0;

Programa 3'

- E0: leia cartão e guarde em E15
- E1: leia cartão e guarde em E14
- E2: multiplique E15 por E14 e guarde em E13
- E3: imprima E13
- E4: pare

Programa 3

- E0:111 00000000 1111
- E1:111 00000000 1110
- E2:101 1111 1100 0110
- ...
- E13:0000000000000000

De 1940 aos dias de hoje

- Os programas eram escritos usando 0's e 1's: Linguagem de Máquina
- Linguagem de Montagem (assembly language) uso de mnemônicos
- Linguagens de Alto nível
- Aperfeiçoamento das linguagens de alto nível, orientação a objetos, aspectos etc