

Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

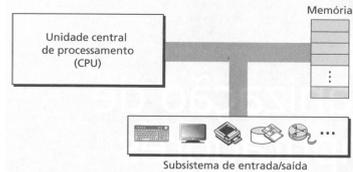
Para você fazer

- Um computador tem palavras de 1 byte e 33554432 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 218 222 237 205 984 658 305 354 927 331 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 230 218 237 203 834 753 232 348 362 231 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

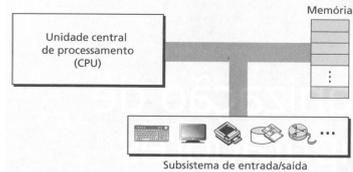
Para você fazer

- Um computador tem palavras de 1 byte e 262144 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 217 222 218 207 373 977 404 211 999 814 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 230 231 217 239 665 566 994 281 674 474 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

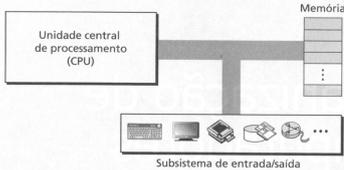
Para você fazer

- Um computador tem palavras de 1 byte e 1048576 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 205 203 235 223 363 920 416 827 273 701 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 230 216 210 222 957 406 251 736 269 568 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina – aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra-rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor.

Instruções de controle de fluxo: Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória.

Instruções de entrada e saída: Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória:
RAM: Random access memory. Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia.
ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode ser dividido em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 < 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 < 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 < 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

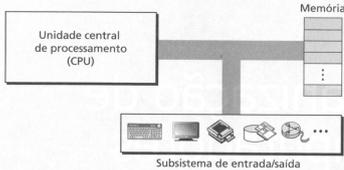
📖 Para você fazer

1. Um computador tem palavras de 1 byte e 2147483648 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
2. Um interpretador do computador acima descrito recebeu o programa 234 215 226 322 220 644 292 675 367 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
3. Um interpretador do computador acima descrito recebeu o programa 213 219 211 235 491 578 812 506 280 748 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina – aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra-rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória.

Instruções de entrada e saída: Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória:
RAM: Random access memory. Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia.
ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode ser dividido em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma sequência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 < 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 < 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 < 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

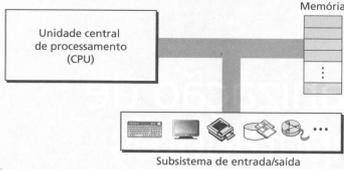
Para você fazer

- Um computador tem palavras de 1 byte e 2097152 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória?
- Um interpretador do computador acima descrito recebeu o programa 212 232 225 236 815 703 250 769 219 412 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 220 218 226 236 726 278 660 243 341 997 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina – aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra-rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor.

Instruções de controle de fluxo: Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória.

Instruções de entrada e saída: Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória:
RAM: Random access memory. Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia.
ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode ser dividido em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

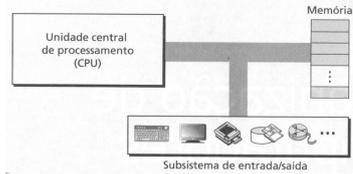
Para você fazer

- Um computador tem palavras de 1 byte e 4294967296 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 217 203 204 230 472 404 947 202 948 516 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 204 210 230 209 599 481 863 685 440 795 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 < 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 < 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 < 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

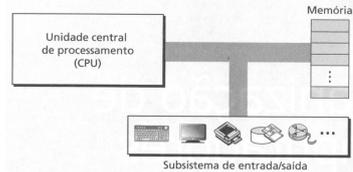
Para você fazer

- Um computador tem palavras de 1 byte e 33554432 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 240 226 231 233 713 524 506 530 926 711 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 212 217 223 233 469 714 982 646 505 890 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

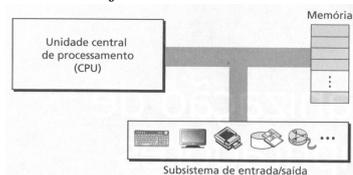
Para você fazer

- Um computador tem palavras de 1 byte e 536870912 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 217 208 212 215 293 662 834 613 357 217 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 223 206 223 233 296 532 297 594 517 846 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

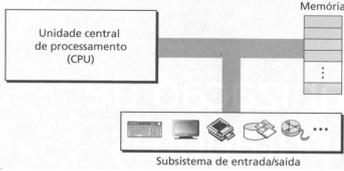
Para você fazer

- Um computador tem palavras de 1 byte e 524288 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 217 213 231 203 839 344 206 803 490 744 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 218 224 228 211 421 739 845 999 630 552 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina – aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra-rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor.

Instruções de controle de fluxo: Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória.

Instruções de entrada e saída: Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória:
RAM: Random access memory. Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia.
ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 < 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 < 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 < 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

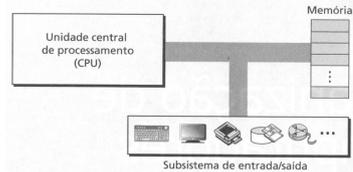
Para você fazer

- Um computador tem palavras de 1 byte e 524288 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 213 237 240 223 997 421 606 928 329 635 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 217 217 225 215 814 839 573 699 782 958 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

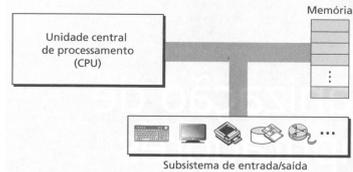
Para você fazer

1. Um computador tem palavras de 1 byte e 4194304 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
2. Um interpretador do computador acima descrito recebeu o programa 216 237 237 240 757 870 779 586 618 476 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
3. Um interpretador do computador acima descrito recebeu o programa 230 239 229 239 782 654 971 324 271 741 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

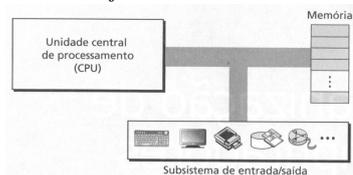
Para você fazer

- Um computador tem palavras de 1 byte e 33554432 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 202 234 234 551 794 268 551 950 856 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 237 215 214 217 613 861 692 336 855 627 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

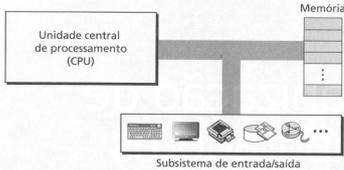
Para você fazer

- Um computador tem palavras de 1 byte e 17179869184 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 223 210 207 208 401 610 414 634 912 782 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 224 209 227 212 204 332 823 267 283 977 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina – aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra-rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor.

Instruções de controle de fluxo: Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória.

Instruções de entrada e saída: Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória:
RAM: Random access memory. Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia.
ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode ser dividido em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 < 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 < 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 < 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

Para você fazer

- Um computador tem palavras de 1 byte e 536870912 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 232 223 218 233 719 407 328 299 996 680 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 225 212 203 225 865 497 233 586 738 383 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

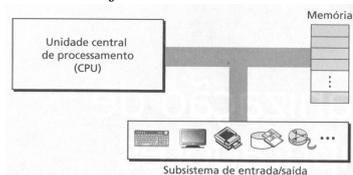
Para você fazer

- Um computador tem palavras de 1 byte e 67108864 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 224 202 239 222 674 234 310 853 266 865 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 218 206 240 208 969 844 374 536 451 810 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

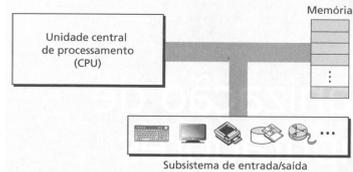
Para você fazer

- Um computador tem palavras de 1 byte e 131072 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 215 217 232 202 263 607 450 506 986 441 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 206 218 215 210 280 340 224 958 497 768 100 Após sua execução qual a soma de $R0+R1+R2+R3$?



Arquitetura de computadores I

Pode-se dividir – grosso modo – um computador em três subsistemas: unidade central de processamento, memória principal e subsistema de entrada e saída. Veja um desenho disso:



Unidade Central de Processamento: Conhecida pela sigla CPU, é o processador principal. É o componente mais caro de um computador. Pode ainda ser subdividido em outros 3 componentes: ULA (unidade lógico-aritmética), unidade de controle e conjunto de registradores. A unidade de controle é responsável por decodificar, interpretar e executar as instruções de máquina (– aqui a grande contribuição de Von Neumann). A unidade aritmético-lógica simplesmente (!) implementa a aritmética que vimos no ensino fundamental para dentro da máquina, de modo milhões ou bilhões de vezes mais rápido que nós lá no fundamental e mesmo hoje. Finalmente, os registradores são localizações de armazenamento ultra- rápido, independentes, para manipulações aritméticas temporárias. Por exemplo, uma operação de máquina pode ler um conteúdo da memória e guardá-lo num registrador determinado. Outra, pode somar dois registradores e uma terceira pode devolver um registrador para dentro da memória.

Eis alguns exemplos de instruções de máquina **Instruções aritméticas e lógicas:** Adição (ADD): soma dois valores e armazena o resultado em um registrador ou na memória. Subtração (SUB): subtrai um valor de outro e armazena o resultado em um registrador ou na memória. Multiplicação (MUL): multiplica dois valores. Divisão (DIV): divide um valor por outro. AND: executa a operação lógica AND bit a bit entre dois valores. OR: executa a operação lógica OR bit a bit entre dois valores. NOT: inverte os bits de um valor. **Instruções de controle de fluxo:** Jump (JMP): transfere o controle da execução para outra instrução em um endereço específico. Conditional Jump (JNZ, JE, JG, etc.): transfere o controle da execução para outra instrução apenas se uma condição específica for verdadeira. Call: chama uma subrotina e salva o endereço de retorno para que a execução possa voltar após a subrotina ser concluída. Return: retorna da subrotina para o local de onde foi chamada.

Instruções de acesso à memória: Load (LOAD): copia um valor da memória para um registrador. Store (STORE): copia um valor de um registrador para a memória. **Instruções de entrada e saída:** Read: lê dados de um dispositivo de entrada e os armazena em um registrador ou na memória. Write: escreve dados em um dispositivo de saída a partir de um registrador ou da memória.

Instruções especiais: Move (MOV): copia um valor de um registrador para outro. Compare (CMP): compara dois valores e define flags de status que indicam o resultado da comparação. Halt: interrompe a execução do programa.

A memória principal consiste de um conjunto de localizações de armazenamento, cada uma com um identificador único chamado *endereço*. Dados são transferidos de e para a memória em grupos de bits chamados *palavras*. Uma palavra pode ser um conjunto de 8, 16, 32, 64 (e crescendo) bits. Por questões históricas, a palavra de 8 bits é chamada *byte* ou em português *octeto*.

Apesar de programadores usarem nomes para identificar trechos de memória, a nível de hardware, o único identificador válido é o endereço. O número total de localizações (endereços) é chamado espaço de endereçamento. Por exemplo, uma memória com 1GByte e palavras iguais a 1 byte, tem um espaço de endereçamento que varia entre 0 e $2^{30} = 1.073.741.824$. Como os computadores operam em padrões binários, isso significa que

neste exemplo serão necessários 30 bits para compor qualquer endereço.

Alguns tipos de memória: **RAM: Random access memory.** Compõe a maior parte da memória e pode ter qualquer endereço lido e/ou gravado. É volátil, o que significa que é estável enquanto houver energia. **ROM: Read-only memory, PROM: Programable ROM, EPROM: Erasable PROM, EEPROM: Electrically EPROM.**

Além destas memórias, costuma-se trabalhar com uma hierarquia de memórias: na base a memória principal (grande e barata), seguida pela memória cache, e no topo os registradores (caros, logo poucos e ultra-rápidos). O conceito de cache é intermediário e se beneficia de um fenômeno chamado localidade de referência temporal/espacial. Esta regra também é conhecida como Princípio de Pareto ou regra 80-20.

O subsistema de entrada e saída contempla a comunicação do computador com o mundo externo. Este subsistema ainda pode se dividir em 2 grupos: aquele sem armazenamento (teclado, vídeo, mouse, impressora, joystick...) e os com armazenamento (discos, SSDs, pendrives...). Outra maneira de olhar para os E/S com armazenamento é como memórias, já que podem armazenar dados para posterior recuperação. Sendo assim, eles podem entrar na parte inferior da hierarquia de memórias, já que não são voláteis e tem custos muito inferiores aos da memória. Os discos magnéticos estão divididos em setores e trilhas. Contam com acesso aleatório. A menor unidade de transferência disco \Leftrightarrow CPU é o bloco. Blocos grandes minimizam o tempo de transferência, mas pioram o desperdício de espaço no disco. A decisão quanto ao tamanho do bloco é tomada em tempo de formatação (que poderia ser comparada ao desenho das marcas de um estacionamento feito antes de ele começar a ser usado). Os SSDs (Solid State Disks) tem mais ou menos as mesmas características dos discos magnéticos, mas por não terem partes móveis estão menos sujeitos a defeitos além de terem tempos de acesso menores.

Há aqui um outro tipo de dispositivo (que rapidamente está perdendo relevância) que são os discos óticos: CD-ROMs, CD-R e os DVDs. Neste caso, o disco é tratado como uma longa tira de informações binárias (tudo a ver com sua origem: a gravação de música).

Interconexão entre subsistemas: Esta abordagem é importante já que os 3 subsistemas (CPU, memória e E/S) precisam trocar informações todo o tempo. A CPU e a memória estão conectadas por 3 barramentos: dados, endereços e controle. O barramento de dados transmite 1 bit por linha e havendo 64 linhas, a transferência se dará à base de 64 bits ao mesmo tempo. O barramento de endereços é usado para acessar a memória. Usa n bits para acessar o endereço 2^n e portanto precisa n linhas. Este valor tem a ver com o espaço de endereçamento. Finalmente o barramento de controle indica o tipo de operação que é desejada a cada ciclo. Novamente deve haver m linhas para comandar 2^m operações distintas. deve-se notar que tanto a memória quanto a CPU são dispositivos 100% eletrônicos, sem nenhum tipo de movimento envolvido.

Já a conexão memória \Leftrightarrow E/S é distinta e sobretudo fortemente dependente de QUAL é o dispositivo de E/S acessado. Para uniformizar tais acessos (que são inerentemente distintos) usa-se o conceito de controlador (ou interface). Aqui reinam o SCSI (Small computer system interface), antigo, mas ainda usado em alguns casos, USB (universal serial bus), que conecta muitas coisas (teclados, mouses, discos, pendrives, câmeras...), passando por SATA (serial ATA, usado em discos rígidos e SSDs) e NVMe (Nonvolatile Memory Express, ostentando velocidade e desempenho excepcionais para SSDs). Este último é uma tecnologia ainda em desenvolvimento. ATA é a sigla *Advanced Technology Attachment*, marca comercial do lançador a AST em 1986.

A USB pode ser *hot-swappable* (=conectado e removido sem ser necessário reinicializar o computador), e admite o conceito de árvore, sendo o controlador imaginado como o nodo raiz. A versão 2.0 do padrão USB admite até 127 nodos nessa árvore. A USB utiliza um cabo de 4 linhas: duas conduzem eletricidade (+5V e terra) para dispositivos de baixa potência. As outras duas são trançadas (para diminuir o ruído) e conduzem dados, endereços e sinais de controle. A interface USB utiliza quatro tipos de conector físico: o retangular A (conecta o

controlador), o quadrado B (conecta o dispositivo), enquanto os mini-A e mini-B conectam dispositivos menores em tamanho. A versão 2.0 define 3 taxas de transferência: 1.5 Mbps, 12 Mbps e 480 Mbps. Os dados são transferidos em pacotes. Já o padrão USB 3.0 apresenta as velocidades de 5, 10 e 20 Gbps.

Interpretador Um certo computador tem 10 registradores e 1000 palavras de RAM. Cada registrador ou cada endereço da RAM pode conter um inteiro de 3 dígitos variando entre 0 e 999. As instruções são codificadas como um inteiro de 3 dígitos e armazenadas em RAM. São elas:

100	pare
2dn	coloque o valor n no registrador d
3dn	adicione n ao registrador d
4dn	multiplique o registrador d por n
5ds	coloque o valor do registrador s no registrador d
6ds	adicione o valor do registrador s no registrador d
7ds	multiplique o registrador d pelo valor do registrador s
8da	coloque em d o valor da RAM cujo endereço está no registrador a
9sa	coloque na RAM (endereço está em a) o valor que está no registrador s
0ds	vá para o endereço que está em d , a menos que o registrador s contenha 0

Todos os registradores inicialmente contém 0. O conteúdo inicial da RAM é lido da entrada. A primeira instrução a ser executada está no endereço 0 da RAM. Todos os resultados devem ser reduzidos ao módulo 1000.

Entrada Cada entrada consiste de uma seqüência de números significando o conteúdo da RAM a partir do endereço 0. Endereços não referidos aqui são inicializados com 0. O sinal de fim de dados é um único 0.

Saída Na maratona original, a saída era o número de instruções executadas (incluindo a instrução PARE). Pode-se assumir que todo programa deverá parar. **Entretanto, nesta folha**, a resposta esperada é a soma dos registradores $R0+R1+R2+R3$. **Exemplo**

entrada
 299 492 495 399 492 495 399 283 279 689 078 100
 230 202 216 207 349 874 287 782 855 863 100
 238 208 222 210 357 496 580 462 303 869 100
 0
 saída
 0, 13, 21

A seguir, uma interpretação dos comandos referentes a primeira linha acima:

000.299=coloque 9 em R9 fica R9=0009
 001.492=multiplique R9 por 2 fica R9=0018
 002.495=multiplique R9 por 5 fica R9=0090
 003.399=some 9 em R9 fica R9=0099
 004.492=multiplique R9 por 2 fica R9=0198
 005.495=multiplique R9 por 5 fica R9=0990
 006.399=some 9 em R9 fica R9=0999
 007.283=coloque 3 em R8 fica R8=0003
 008.279=coloque 9 em R7 fica R7=0009
 009.689=some a R8 o valor de R9 fica R8=0002
 010.078=Se R8 <> 0, va para R7 R8=0002 e R7=0009
 011.689=some a R8 o valor de R9 fica R8=0001
 012.078=Se R8 <> 0, va para R7 R8=0001 e R7=0009
 013.689=some a R8 o valor de R9 fica R8=0000
 014.078=Se R8 <> 0, va para R7 R8=0000 e R7=0009
 015.100=Parando...

Para você fazer

- Um computador tem palavras de 1 byte e 536870912 bytes de memória. Quantos bits são necessários para endereçar qualquer byte na memória ?
- Um interpretador do computador acima descrito recebeu o programa 215 223 204 217 571 408 661 236 644 783 100 Após sua execução qual a soma de $R0+R1+R2+R3$?
- Um interpretador do computador acima descrito recebeu o programa 219 215 238 231 683 226 761 707 320 522 100 Após sua execução qual a soma de $R0+R1+R2+R3$?

