**Try your best!☘**

**\_\_Ocean\_\_**

**30 câu lý thuyết, 5 câu bài tập: điền đáp án**

**Chương 1.**

1. ***Hệ đếm:***

***khái niệm, phân biệt hệ đếm có vị trí/ko có vị trí, tính giá trị số, trọng số của các chữ số***

* Khái niệm:
* Hệ đếm la một tập các kí hiệu chữ số để biểu diễn các số và xác định giá trị của các biểu diễn số.
* Hệ đếm không có vị trí:
* Mỗi kí hiệu biểu diễn một giá trị nhất định, không phụ thuộc vào vị trí đứng: VD số la mã,…
* Hệ đếm có vị trí
* Giá trị của số phụ thuộc vào vị trí của các phần tử: VD hệ 2, hệ 10, hệ 8, hệ 16, ….
* **Cơ số** của hệ đếm r là số ký hiệu đc dùng: VD hệ cơ số 8 có cơ số là 8 (0,1,2,3,4,5,6,7)
* **Trọng số** = r^i (i là số int âm/dương) giúp phân biệt chữ số khác nhau. VD số **1**001(2) số ở vị trí i=3 có trọng số là 2^3
* [ClickToVideo](https://www.youtube.com/watch?v=Xkr_GhrxvE4)

Graphical user interface, application

Description automatically generated

***Chuyển đổi hệ đếm:***

***Click*** [Công cụ tính toán](https://www.rapidtables.com/convert/number/octal-to-hex.html?x=524.7)

* ***Nhị phân 🡪 Thập phân 🡪Nhị phân***

VD; 11010001(2) = 1x2^7 + 1x2^6 + 1x2^4 + 1x2^0 = 128+64+16+1 = 209(10)

VD: 123(10) = 1111011(2) (chia liên tiếp cho 2, lấy số dư từ dưới lên)

* ***Nhị phân 🡪 Bát phân 🡪 Nhị phân***

Table

Description automatically generated VD; 11010001.010(2) = 011 010 001 . 010 = 3 2 1 . 2 = 321.2(8)

VD: 120(8) = 1 2 0 = 001 010 000 = 001010000(2)

321.2(8) = 3 2 1.2 = 011 010 001.010 = 011010001.010(2)

* ***Nhị phân 🡪 Thập lục phân 🡪 Nhị phân***

Table

Description automatically generatedVD 11010001.001(2) = 1101 0001.0010 = D 1. 2 = D1.2(16)

VD: 2D4(16) = 2 D 4 = 0010 1101 0100(2)

3B4.A(16) = 3 B 4.A = 0011 1011 0100.1010

=

* ***Bát phân 🡪 Thập phân 🡪 Bát phân:***

VD: 534(8) = 5x8^2 + 3x8^1 + 4x8^0 = 320 + 24 + 4 = 348(10)

430(10) = 656(8) (chia liên tiếp cho 8, lấy số dư từ dưới lên)

* ***Bát phân 🡪 Thập lục phân 🡪 Bát phân***

VD: 652(8) = 6 5 2 = 110 101 010 = 1 1010 1010 = 1AA

524.7(8) = 5 2 4.7 = 101 010 100.111 = 1 0101 0100.1110 = 154.E

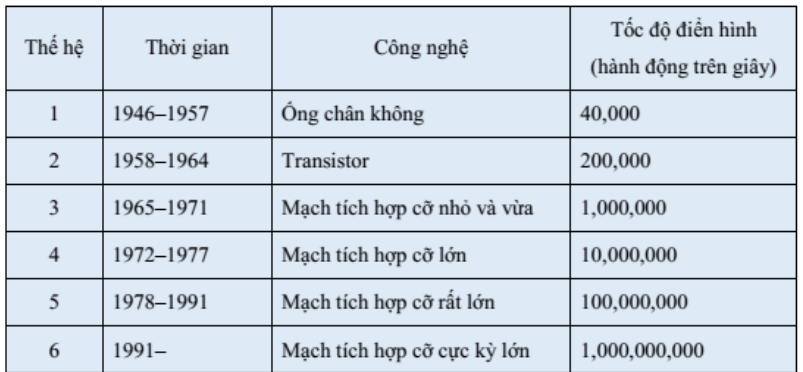
1. ***Cấu trúc (KTMT), chức năng của máy tính (4)***

* Cấu trúc máy tính:
* CPU: Bộ xử lý trung tâm
* Bộ điều khiển CU: điều khiển hoạt động của CPU và máy tính
* Bộ làm toán và luận lý ALU: thực hiện chức năng xử lý dữ liệu và tính toán
* Thanh ghi Registers: cung cấp lưu trữ nội bộ cho CPU
* Các kết nối trong CPU: cung cấp thông tin liên lạc giữa các khối CU, ALU, thanh ghi
* Bộ nhớ chính (ROM, RAM): Là tập hợp các ô nhớ lưu trữ dữ liệu đc mã hóa số nhị phân.
* Bus: thành phần trung gian truyền thông tin giữa CPU-Bộ nhớ trong(chính)-I/O
* Module vào/ra (I/O): bộ phận nhập xuất thông tin
* Chức năng máy tính (4):
* Xử lý dữ liệu (data processing)
* Lưu trữ dữ liệu (Data storage)
* Di chuyển dữ liệu(Data movement)
* Điều khiển (control)

**Chương 2**. ***Lịch sử: máy tính đầu tiên, hệ đếm đầu tiên***

***Các giai đoạn của máy tính: 6 thế hệ: công nghệ phần cứng (đèn ống chân không, transistor, vi mạch IC – mạch tích hợp)***

* Lịch sử:
* Máy tính đầu tiên ENIAC(1943-1946):
* Cha đẻ: John Mauchly và học trò John Eckert
* Nhiệm vụ đầu tiên là thực hiện một loạt các tính toán để xác định tính khả thi của bomb hydrogen
* EDVAC(1945)
* Cha đẻ: John Von Neumann
* C.trình biểu diễn dưới dạng thích hợp để lưu vào bộ nhớ cùng với data
* IAS(1952)
* Là nền tảng của máy tính hiện đại ngày nay
* Có các thanh ghi:
* Thanh ghi bộ nhớ đệm – MBR: Chứa dữ liệu đc ghi vào từ bộ nhớ or nhận data đọc từ bộ nhớ
* Thanh ghi địa chỉ bộ nhớ - MAR: Chứa địa chỉ trong bộ nhớ cho lần đọc/ghi tiếp theo
* Thanh ghi tập lệnh – IR: Chứa lệnh đang thực thi
* Thanh ghi IBR: lưu trữ lệnh nằm bên tay phải của 1 từ (word) trong bộ nhớ.
* Thanh ghi PC: chứa địa chỉ lệnh tiếp theo
* Bộ cộng tích lũy AC, bộ nhân chia MQ: lưu giữ tạm thời các toán hạng và kết quả của các phép tính trong ALU
* UNIVAC(1947)
* Máy tính thương mại thành công đầu tiên
* UNIVAC II hoàn thành vào cuối 1950
* Hệ đếm đầu tiên của máy tính là hệ **Thập phân**
* Các thế hệ máy tính: 6 thế hệ



* Cấu tạo của Vi mạch - IC:
* Gate – phần tử logic: thực hiện các phép toán logic và Boolean đơn giản, truyền data
* Cell – Ô nhớ: Lưu trữ các giá trị 0 or 1
* Một chip/IC bao gồm nhiều gate và cell
* Chip được đóng gói
* Số lượng transistor trên mỗi chip sẽ tăng x2 sau mỗi năm (1965) vs giá thành ko đổi, sau năm 1970 thì x2 sau mỗi 18 tháng

***Vi mạch: gate (phần tử logic)/cell (ô nhớ)***

**Chương 3.**

1. ***Hoạt động máy tính: chu kỳ lệnh: truy xuất, thưc thi; các thanh ghi và ý nghĩa (điều khiển: PC, MAR, …)***

* Chu kì lệnh: quà trình CPU thực thi c.trình là khi CPU liên tục thực hiện các chu kì lệnh
* Chu kì lệnh gồm:
* Truy xuất lệnh: Lấy lệnh từ RAM 🡪 CPU
* Thực thi lệnh: các bước tùy thuộc vào từng lệnh
* Chu kỳ truy xuất:
* Tính toán địa chỉ: PC nạp địa chỉ lệnh vào CPU
* Truy xuất lệnh: MAR nạp địa chỉ vào Memory 🡪 đọc giá trị tại MAR (M[MAR]) nạp vào MBR 🡪 MBR nạp vào IR
* Giải mã lệnh: CU thực hiện giải mã lệnh trong IR
* Chu kỳ thực thi lệnh:
* Tính toán địa chỉ toán hạng: chuyển địa chỉ toán hạng vào MAR
* Truy xuất toán hạng
* Tính toán dữ liệu: ALU: MBR + AC 🡪 AC
* Tính toán địa chỉ lưu trữ
* Lưu toán hạng vào bộ nhớ.
* Các thanh ghi :
* Thanh ghi bộ nhớ đệm – MBR: Chứa dữ liệu đc ghi vào từ bộ nhớ or nhận data đọc từ bộ nhớ
* Thanh ghi địa chỉ bộ nhớ - MAR: Chứa địa chỉ trong bộ nhớ cho lần đọc/ghi tiếp theo
* Thanh ghi tập lệnh – IR: Chứa lệnh đang thực thi
* Thanh ghi I/O AR: xác định một thiết bị I/O cụ thể
* Thanh ghi I/O BR: Trao đổi data: module I/O vs CPU
* Thanh ghi PC: chứa địa chỉ lệnh tiếp theo

1. ***Hệ thống bus: điều khiển, dữ liệu (hiệu suất ht), địa chỉ (dung lượng bộ nhớ tối đa VXL quản lý đc), bus chuyên dụng, trọng tài bus, truyền dữ liệu đồng bộ/không đồng bộ***

* Bus dữ liệu: truyền dữ liệu
* Có các loại 32 đường, 64 đường, 128 đường…
* Số lượng đường = độ rộng bus 🡪 yếu tố chính quyết định hiệu suất toàn system
* Mỗi đường chỉ truyền 1b/lượt 🡪 Số lượng đương = số lượng bit đc truyền cùng một lúc
* Bus địa chỉ: xác định vị trí ngăn nhớ (M), thiết bị ngoại vi
* Độ rộng bus = dung lượng nhớ max của system
* Vs **n** đường 🡪 có **2^n** ngăn nhớ 🡪 **2^n-1** (ko gian chỉ bộ nhớ) giá trị địa chỉ bộ nhớ
* Bus điểu khiển: truyền các tín hiệu điều khiển đọc/ghi
* Tín hiệu yo cầu ngắt (M I/O 🡪 CPU)
* Các tín hiệu điều khiển cả thông tin lệnh và định thời giữa các module system
* Bus chuyên dụng: (bus dữ liệu, bus địa chỉ, bus điều khiển)
* Sử dụng vs 1 chức năng cụ thể
* Ưu điểm: Nhanh hơn, ít xung đột
* Nhược điểm: Tăng kích thước và chi phí
* Bus ghép kênh: (Thông tin = data + address)
* Các thông tin đc truyền trên cùng một đường
* Ưu điểm: Ít đường hơn, tiết kiệm chi phí
* Nhược điểm: mạch phức tạp hơn
* Trọng tài bus: Xác định thiết bi ưu tiên truyền bus
* Phân sử tập trung
* Phân sử phân tán
* Truyền dữ liệu đồng bộ:
* Các h.động truyền trên bus được thực hiện theo xung đồng hồ
* Đường xung đồng hồ truyền liên tiếp các bit 0 vs 1
* Thực hiện truyền đơn giản nhưng ít linh hoạt hơn ko đồng bộ
* Truyền dữ liệu không đồng bộ:
* Không sử dụng tín hiệu đồng hồ
* Data 🡪 Bus 🡪 memory gửi tín hiệu ACK 🡪 VXL 🡪VXL thực hiện đọc/ghi data

***Làm lại các bài tập trong slide***

**Chương 4.**

1. ***Tổng quan bộ nhớ máy tính: đơn vị dữ liệu BN: từ nhớ/khối nhớ***

***Phương pháp truy cập***

***Tham số hiệu năng***

***Khả biến (điện động)/bất khả biến (điện tĩnh)***

* Tổng quan về bộ nhớ máy tính
* Từ (word):
* Là đơn vị tự nhiên của bộ nhớ
* Kích thước từ = số bit biểu diễn 1 số nguyên và kích thước lệnh

VD: intel x86 có kích thước từ là 32b

* Đơn vị đánh địa chỉ là byte or word: 2^A = N (N là đơn bị đánh địa chỉ; A số bit địa chỉ)
* Đơn vị truyền:
* Bộ nhớ chính: đơn vị truyền = số bit đc gửi đến/đi từ bộ nhớ
* Bộ nhớ ngoài: đơn vị truyền là các khối (block)
* Phương pháp truy cập:
* Truy cập tuần tự - Băng từ: Cấu thành từ các bản ghi (Record)
* Truy cập trực tiếp – Đĩa từ: Mỗi khối/bản ghi có một địa chỉ duy nhất dựa trên vị trí vật lý
* Truy cập ngẫu nhiên – Memory: Các vị trí trong ô nhớ có một cơ chế định địa chỉ riêng
* Kết hợp – Cache: Các từ được truy xuất dựa vào nội dung thay vì vị trí
* Tham số hiệu năng:
* 2 đặc điểm quan trọng nhất của bộ nhớ là dung lượng và hiệu năng
* Ba tham số hiệ năng được sử dụng là:
* Thời gian truy cập (độ trễ)
* Chu kỳ bộ nhớ
* Tốc độ truyền tải
* Bộ nhớ khả biến (điện động): thông tin bi suy yếu or mất đi khi ko còn nguồn điện
* Bộ nhớ bất khả biến (điện tĩnh): Thông tin đã đc ghi thì sẽ không bị mất đi khi ko có nguồn điện cung cấp

1. ***Bộ nhớ cache:***

***Nguyên lý***

***Yếu tố thiết kế cache: cache vật lý / cache logic (ảo)***

***Ánh xạ bộ nhớ: 3 phương pháp, các thực hiện, xác địn dữ liệu, tính toán các trường của phươg pháp ánh xạ, thuật toán thay thế, chính sách ghi,…***

* Nguyên lý bộ nhớ cache
* BN chính gồm 2^n word được đánh địa chỉ: n bit địa chỉ
* BN chính gồm các block: 1 block = K word
* BN chính có 2^n/K = M block
* BN cache gồm các line, mỗi line có K word
* Mỗi block của BN chính 🡪 ánh xạ vào 1 line của cache
* Bộ XL muốn đọc 1 word 🡪 kiểm tra từ có trong cache ko
* Có: word 🡪 VXL
* Không: một block của BN chính (bao gồm tư bộ XL muốn lấy) đc đọc vào cache 🡪 word đc gửi đến VXL
* Yếu tố thiết kế cache
* Cache vật lý: BN cache được đạt giữa MMU và BN chính, sử dụng địa chỉ vật lý.
* Cache logic(ảo): BN cache đặt giữa BXL và MMU, sử dụng đỉa chỉ ảo
* Ánh xạ bộ nhớ:
* Trực tiếp:
* Mỗi block 🡪 1 line duy nhất của cache
* Đơn giản nhất
* Logic cache tách địa chỉ BN thành 3 trường:
* Word (w bit): xác định một word trong block
* Line (r bit) xác định block đó đc ánh xạ vào line thứ bnhieu trong cache
* Tag(s-r bit) xác định block nào đang được ánh xạ vào line đó
* Độ dài địa chỉ = s + w bit
* Số ô nhớ trong bộ nhớ chính = 2^(s+w) word | byte
* Kích thước khối = kích thước line = 2^w word | byte
* Số block trong BN chính = 2^(s+w)/2^w = 2^s
* Số đường trong bộ nhớ cache = m = 2^r
* Kích thước của Tag = s – r bit
* Kết hợp
* Mỗi block 🡪 1 đường bất kì của cache
* Logic điều khiển cache = Tag + Word
* Để xác định 1 khối có trong cache ko? Ta kiểm tra trường TAG của all line
* Khắc phục nhược điểm của Trực tiếp
* Logic cache tách địa chỉ BN thành 2 trường:
* Word(w bit): xác đinh 1 word trong block
* Tag(s bit): xác định block nào đang đc ánh xạ vào line đó
* Chiều dài địa chỉ = s + w bit
* Số ô nhớ đc đánh địa chỉ = 2^(s+w) word | byte
* Kích thước khối = kích thước line = 2^w word | byte
* Số block trong BN chính = 2^(s+w)/2^w = 2^s
* Số đường trong cache = ko xác định
* Chiều dài trường Tag = s bit
* Tập kết hợp
* Tập kết hợp = Trực tiếp + Kết hợp
* Chia cache thành một số Tập (set), mỗi tập chứa một line
* m = v\*k
* i = j/v (lấy dư)

+) i = STT Tập trong cache

+) j = STT block trong BN chính

+) m = số line trong cache

+) v = số Tập trong cache

+) k= số lượng line trong mỗi tập

* Chiều dài địa chỉ = s + w bit
* Số ô nhớ đc đánh địa chỉ = 2^(s+w) word | byte
* Kích thước khối = kích thước line = 2^w word | byte
* Số block trong BN chính = 2^(s+w)/2^w = 2^s
* Số lượng đường trong 1 tâp = k
* Số lượng tập = v = 2^d
* Số lượng đường trong cache = m = k\*v = m\*2^d
* Kích thước cache = k\*2^(d+w) word | byte
* Độ rộng trường Tag = s – d bit
* Thuật toán thay thế
* LRU
* Hiệu quả nhất, phổ biến nhất
* Thay thế khối trong cache lâu nhất mà ko tham chiếu đến nó
* FIFO
* Thay thế khối đã nằm trong cache lâu nhât
* Dễ dàng thực hiện như một ký thuậ vòng đệm or round-robin
* LFU
* Thay thế khối có it tham chiếu tơi nó nhất
* Mỗi khi tham chiếu đến line nào BN của line đó +1 đơn vị
* Ngẫu nhiên
* Có thể thay thế bất cứ khối nào
* Có thể gây giảm hiệu suất của system một chút
* Chính sách ghi
* Write through
* Write back

**Chương 5:** **ROM/RAM**

1. ***RAM: SRAM, DRAM***

* SRAM-RAM tĩnh
* SD các thành phần giống nhau trong bộ XL
* Các giá trị nhị phân đc lưu trong các cổng logic flip-flop truyền thống
* DRAM – RAM động
* Các ô lưu trữ data bằng cách nạp cho các tụ điện
* Điện tích có or ko trên tụ điện ứng vs bit 1 vs 0
* Dòng làm tươi(định kỳ) khôi phục dòng điện bị rò rỉ trong quá trình truyền data

|  |  |
| --- | --- |
| **DRAM** | **SRAM** |
| * Tốc độ đọc/ghi nhanh * Đều là bộ nhớ điện động * Sử dụng cho bộ nhớ trong | |
| - Lưu trữ data trên tụ điện 🡪 Tốc độ đọc/ghi chậm hơn  - Giá thành rẻ hơn  - Mạch đơn giản 🡪 mật độ cell/đơn vị S lớn  - Yêu cầu dòng làm tươi  - Phù hợp bộ nhớ dung lượng cao | - Tốc độ nhanh hơn  - Đắt hơn  - Mạch phức tạp 🡪 mật độ cell/đơn vị S nhỏ  - Phù hợp bộ nhớ dung lượng nhỏ, yêu cầu tốc độ cao |

1. ***ROM: 5 loại***

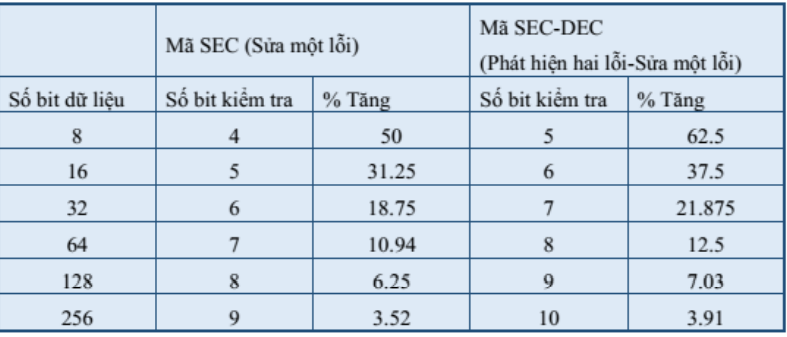
* ROM:
* Chứa data vĩnh cửu ko thể thay thế or thêm vào
* Ko cần nguồn điện để duy trì
* Nhược điểm:
* Ko có chỗ cho lỗi 🡪 nếu sai 1 bit sẽ mất cả ROM
* Việc nạp data vào ROM mất 1 khoảng time và chi phí khá lớn
* PROM:
* Thay thế ít tốn kém
* Ko xóa đc và chỉ ghi đc 1 lần duy nhất
* Quá trình ghi data đc thưc hiện = điện và phải dùng thiết bị đặc biệt
* Linh hoạt và tiện lợi
* EPROM
* Có thể xóa đc và xóa đc nhiều lần, xóa bằng tia UV
* Đắt hơn PROM
* EEPROM
* Có thể xóa đc bằng điện
* Có thể ghi mà ko cần xóa data trước đó
* Đắt hơn EPROM
* Flash Memory
* Trung gian giữa EPROM và EEPROM
* Sử dụng công nghệ xóa điện, ko cho phép xóa cấp độ byte, chỉ xóa ở cấp đọ block
* Tốc độ nhanh hơn
* Mỗi cell chỉ sd 1 transistor, mật độ cell lớn hơn các loại trên

1. ***Tổ chức chip: chân tín hiệu, tổ chức chip,…***

* Các đường địa chỉ: có 2^n từ nhớ
* Các đường data: Độ dài từ nhớ = m bit
* Dung lượng chip nhớ = 2^n x m bit
* Các đường điểu khiển
* Tín hiệu chọn chip CS: Chip Select
* Tín hiệu điều khiển đọc OE: Output Enable
* Tín hiệu điều khiển ghi WE :Write Enable

1. ***Mã SEC, SEC-DED: cách tính, số lượng bit mã, ý nghĩa mã,…***

* Khi đọc bộ nhớ, sử dụng hàm f tính lại mã trên dữ liệu lấy ra , so sánh với K bit mã lưu trữ. 3 trường hợp xảy ra:
* Không phát hiện ra lỗi. Dữ liệu được gửi đi
* Phát hiện ra lỗi, có thể sửa lỗi. Dữ liệu và bit sửa lỗi được đưa vào bộ sửa lỗi sau đó được gửi đi
* Một lỗi được phát hiện nhưng không thể sửa. Lỗi sẽ được báo cáo
* Mã này được gọi là mã CRC
* Số lượng bit ktra:



* Bố cục các bit data và các bit ktra

Table

Description automatically generated

* SEC-DEC:
* Mã phát hiện 2 lỗi, sửa 1 lỗi
* Sử dụng thêm 1 bit parity

**Chương 6: Bộ nhớ ngoài:**

1. ***HDD: nguyên lý: bộ nhớ từ, cấu trúc và tổ chức dữ liệu (track, sector, cylinder, …)***

* Dữ liệu đc bố trí thành cá vòng trên platter gọi là Track. Độ rộng của track = độ rộng của head
* Các track đc ngăn cách với nhau bởi Gap
* Các track đc chia thành các Sector và chúng đc ngăn cách nhau bởi Gap
* Sector có kích thước cố định or ko cố định. Thông thường sector có kích thước cố định là 512 byte
* Cylinder là tập các head có cơ chế di chuyển cố định cùng nhau, tại cùng một thời điểm các head sẽ đc đặt vào các track có cùng khoảng cách vs tấm đĩa.

1. ***SDD: ưu điểm/nhược điểm, vấn đề tồn tại cần giải quyết***

* Ưu điểm:
* Số thao tác đọc ghi /giây cao hơn
* Tiêu thụ ít năng lượng hơn
* Chạy êm hơn, mát hơn
* Time truy cập ngắn hơn, time trễ ít hơn
* Nhược điểm:
* Giá thành cao
* Tuổi thọ ngắn hơn HDD
* Vấn đề tồn tại
* Hiệu năng của SSD giảm dần khi thiết bi đc sử dụng
* Flash Memory ko thể sử dụng đc sau một số lần ghi (thông thường là 100.000 lần)
* Kỹ thuật kéo dài tuổi thọ:
* Sử dụng cache để giữ chậm và gộp các h.đ ghi
* Dùng thuật toán wear-leveling: phân bố đều các lần ghi lên các block
* Quản lý các bad-block

**Chương 9**. Các loại biểu diễn số

1. ***Số nguyên: ko dấu, dấu-độ lớn, bù 2: miền giá trị, mở rộng phạm vi, tính giá trị biểu diễn, các phép toán (bù 2): lấy âm, …***

**Số 0(+) vs số 1(-)**

* Số nguyên không dấu
* Cho số dạng thập phân 🡪 hệ 2: 67(10) 🡪 100 0011(2) (chia 2 lấy dư từ dưới lên
* Chỉ biểu diễn số dương
* Dấu - độ lớn:
* Là dạng biểu diễn đơn giản nhất

0000 0000 (+0) vs 1000 0000 (-0)

* Có 2 dạng biều diễn số 0:
* Miền giá trị: -[2^(k-1) – 1] 🡪 2^(k-1)-1
* Mở rộng phạm vi: Di chuyển bit dấu ra vị trí ngoài cùng bên trái, sau đó điền m-n bit 0 vào sau bit dấu
* Biểu diễn bù 2
* Chỉ có 1 dạng biểu diễn số 0: 0000 0000
* Cách biểu diễn số dương: Dấu độ lớn ⬄ Bù 2
* Cách biểu diễn số âm: Dấu độ lớn # bù 2
* B1: Lấy bù 1 của số dương tương ứng ⬄ Đảo bit 0🡪1, 1🡪0
* B2: Cộng thêm 1
* Cách 2: Đọc ngược từ dưới lên, gặp bit 1 đầu tiên giữ nguyên, các bit tiếp theo đảo bit
* Miền giá trị có k bit: -2^(k-1) -> 2^(k-1) – 1
* Mở rộng phạm vi: Di chuyển bit dấu sang vị trí ngoài cùng và điền vào sau bit dấu:
* Điền 0 nếu là số dương
* Điền 1 nếu là số âm
* Các phép toán vs Bù 2
* Phép đổi dấu
* B1: Đảo all bit
* B2: Cộng 1
* Phép cộng

***Table

Description automatically generated***

* Phép trừ
* Lấy bù 2 (đảo) của số trừ và cộng vs số bị trừ

A screenshot of a computer

Description automatically generated with low confidence

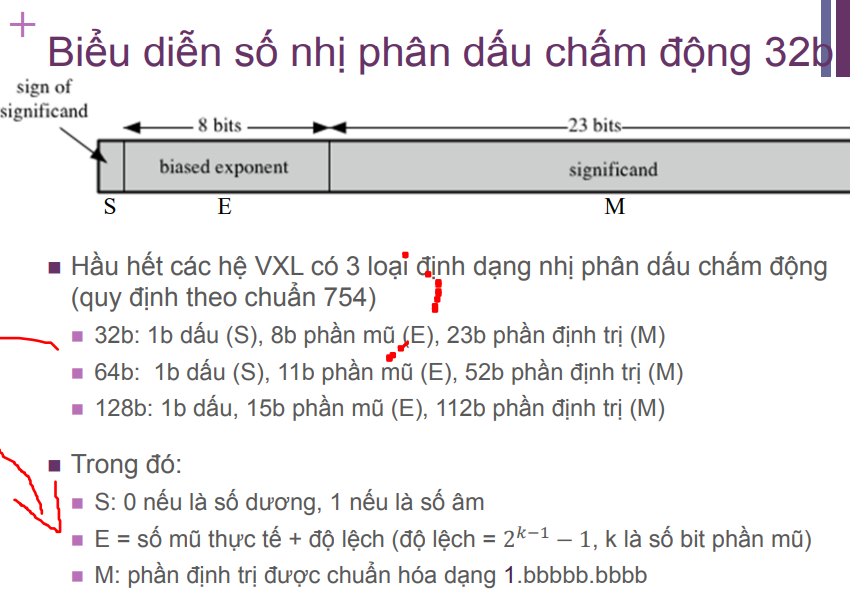
1. ***Số thực dấu chấm động: 32b, 64b, 128b, nguyên tắc phép toán.***

* Một số nhị phân có thể biểu diễn:

***A picture containing text, weapon

Description automatically generated***

* S: phần định trị có dạng 1.bbbb
* B: cơ số, đối vs nhị phân thì mặc định B = 2
* E: Số mũ thực tế
* ***VD*** Vs bộ 32b:
* 1 bit dấu
* 8 bit: số mũ lệch
* E = số mũ lệch – độ lệch
* Độ lệch =  k: số bit phần mũ)
* 23 bit còn lại là phần định trị được quy ước dạng 1.bbbbb…. với số 1 đầu tiên là mặc định

******

* **Nguyên tắc phép toán:**

Text

Description automatically generated

Text

Description automatically generated

**Chương 10:** ***khái niệm tập lệnh, các loại lệnh, lệnh 1 đc/ 3đc,…***

* **Tập lệnh**:
* Là tập hợp các lệnh khác nhau của VXL có thể thực hiện đc
* Mỗi lệnh phải có những thông tin cần thiết cho bộ XL thực hiện h.đ
* H.đ của VXL đc quyết định bởi các lệnh máy tính nó thực hiện
* **Các loại lệnh**
* Xử lý data: các lệnh số học và logic
* Lưu data: Các lệnh đọc/ghi data từ/vào thanh ghi or bộ nhớ
* Di chuyển data: Gồm các lệnh vào/ra
* Điều khiển: gồm các lệnh kiểm tra và rẽ nhánh
* Mã lệnh: Operation code: chỉ ra h.đ đc thực hiện thông qua một mã nhị phân đc gọi là mã lệnh
* Tham chiếu toán hạng nguồn
* Toán hạng đích (tham chiếu toán hạng kết quả)
* **Lệnh 1 đc/2 đc/ 3 đc**
* **Lệnh 0 địa chỉ**
* Ngầm định 2 ngăn nhớ ở đỉnh vùng ngăn xếp (Stack) của BN
* **Lệnh 1 địa chỉ:**
* Một toán hạng ngầm định là thanh ghi AC
* Phổ biến ở các hệ VXL đơn giản, đời đầu
* **Lệnh 2 địa chỉ:**
* Kết quả phép toán đc lưu vào 1 địa chỉ
* **Lệnh 3 địa chỉ:**
* Kích thước lệnh dài
* Toán hạng 1, toán hạng 2, kết quả
* T: vị trí BN tạm thời để lưu trữ kết quả

**🡪** Số lượng địa chỉ tròn mỗi lệnh là một yếu tố cơ bản đối vs thiết kế VXL

🡪 Lệnh càng ít địa chỉ 🡪 kích thước lệnh ngắn hơn 🡪 VXL ít phức tạp hơn 🡪 chương trình

cần nhiều lệnh hơn để thực hiện một công việc 🡪 mất thời gian hơn

**Chương 11:** ***Chế độ địa chỉ: 6 chế độ chính, ý nghĩa, vị trí toán hạng***

* **Tức thì**
* Dạng đơn giản nhất
* Toán hạng = A
* Ưu điểm: chỉ cần truy xuất bộ nhớ 1 lần để lấy lệnh, 🡪 tiết kiệm một chu kỳ cache or bộ nhớ trong chu kỳ lệnh
* Nhược điểm: Kích thước của số bị giới hạn bởi kích thước của trường địa chỉ vì thông thường kích thước của trường này nhỏ hơn kích thước từ
* **Trực tiếp**
* Thuật toán: EA = A
* Trường địa chỉ chứa địa chỉ hiệu dụng của toán hạng
* Ưu điểm: Đơn giản
* Nhược điểm: Không gian địa chỉ hạn chế
* **Gián tiếp**
* Thuật toán: EA = (A)
* Tham chiếu tới địa chỉ của 1 từ trong bộ nhớ chứa địa chỉ đầy đủ của toán hạng
* Ưu điểm: Không gian địa chỉ lớn
* Nhược điểm: Tham chiếu bộ nhớ nhiều lần
* **Thanh ghi**
* Thuật toán: EA = R
* Trường địa chỉ dùng để tham chiếu thanh ghi chứ không phải địa chỉ bộ nhớ chính
* Ưu điểm: Không cần tham chiếu bộ nhớ
* Nhược điểm: Không gian địa chỉ bộ nhớ
* **Gian tiếp thanh ghi**
* Thuật toán: EA = A + ( R )
* Ưu điểm: Linh hoạt
* Nhược điểm: Phức tạp
* **Ngăn xếp**
* Thuật toán: EA = đỉnh ngăn xếp
* Một ngăn xếp là một mảng liên tiếp các ô nhớ: danh sách dạng vào trước ra sau
* Ưu điểm: không cần tham chiếu bộ nhớ
* Nhược điểm: Khả năng ứng dụng ít

**Chương 12:** ***thực thi lệnh, cấu trúc vxl, thanh ghi, đường ống lệnh***

* **Chu kỳ lệnh**
* Truy xuất: đọc lệnh tiếp theo từ bộ nhớ vào bộ VXL
* Thực thi: Giải mã opcode và thực hiện hoạt động đc chỉ định
* Ngắt: VXL lưu lại trạng thái hiện tại của c.trình và chuyển sang phục vụ ngắt
* **Cấu trúc VXL**
* Các yêu cầu xử lý
* Truy xuất lệnh: Bộ xử lý đọc lệnh từ bộ nhớ (thanh ghi, bộ nhớ cache, bộ nhớ chính).
* Giải mã lệnh: Lệnh được giải mã để xác định hành động nào được yêu cầu.
* Truy xuất data: Việc thực thi một lệnh có thể yêu cầu đọc dữ liệu từ bộ nhớ hoặc một module vào/ra
* Xử lý data: Việc thực thi một lệnh có thể yêu cầu thực hiện một số phép toán số học hoặc logic trên dữ liệu.
* Ghi data: Kết thúc việc thực hiện có thể yêu cầu ghi dữ liệu vào bộ nhớ hoặc một module vào/ra
* Tổ chức VXL:
* ALU: khối tính toán số và logic
* CU: Khối điều khiển
* Các thanh ghi: Lưu trữ data tạm thời trong quá trình lệnh đc thực hiện
* **Thanh ghi**
* Các thanh ghi là 1 loại bộ nhớ
* Thanh ghi hiển thị vs ng dùng
* Cho phép ng lập trình n.ngữ assembly or n.ngữ máy sử dụng trong các câu lệnh
* Giảm thiểu các tham chiếu bộ nhớ chính bằng cách sử dụng thanh ghi
* Thanh ghi đa năg
* Thanh ghi data
* Thanh ghi địa chỉ
* Mã điều kiện
* Thanh ghi điền khiển và trạng thái
* Được sử dụng bởi CU để điều khiển hoạt động của bộ vi xử lý và bởi các chương trình hệ điều hành được đặc quyền (privileged) để kiểm soát việc thực thi chương trình
* Thanh ghi PC – bộ đếm chương trình
* Thanh ghi IR – thanh ghi lệnh
* Thanh ghi MAR – thanh ghi địa chỉ bộ nhớ
* Thanh ghi MBR – thanh ghi đệm bộ nhớ
* **Đường ống lệnh**
* Chu kỳ lệnh có 6 giai đoạn:
* Truy xuất lệnh – FI: đọc lệnh tiếp theo vào bộ đệm
* Giải mã lệnh – DI: giải mã opcode và nhận diện toán hạng
* Tính toán địa chỉ toán hạnh – CO: Tính toán địa chỉ hiệu dụng của từng toán hạng nguồn: địa chỉ dịch chuyển, gián tiếp thanh ghi, gián tiếp .v.v....
* Truy xuất toán hạng – FO: Truy xuất từng toán hạng từ bộ nhớ. Không cần truy xuất toán hạng từ thanh ghi
* Thực thi lệnh – EI: : Thực hiện hành động và lưu trữ kết quả (nếu có) trong vị trí toán hạng đích đã định
* Ghi toán hạng – WO: lưu kết quả vào bộ nhớ
* Xung đột trong đường ống lệnh:
* Xung đột tài nguyên: do nhiều công đoạn dùng chung một tài nguyên
* Xung đột data: lệnh sau sử dụng kết quả của lệnh trước
* Xung đột điều khiển: do rẽ nhánh gây ra

**Bài tập:**

1. ***Chuyển đổi hệ đếm***
2. ***Tính toán dung lượng bộ nhớ: RAM, ổ cứng HDD***
3. ***Tính toán định chỉ BN cache theo các phương pháp ánh xạ***
4. ***Tính mã SEC, SEC-DED***
5. ***Phép toán số bù 2***
6. ***Biểu diễn số dấu chấm động***

***­­*\_\_Một số bài tập\_\_**

**CHƯƠNG 8 – HỆ THỐNG SỐ**

**1**. Tổng quát về biểu diễn số trong hệ các hệ đếm.

**2**. Chuyển đổi giữa các hệ đếm (hệ 2, hệ 10, hệ 16).

**3**. Đổi các số thập phân sau sang số nhị phân: 14; 189.

**4.** Đổi các số nhị phân sau sang số thập phân: 10110;10011011.

**5**. Giá trị thập phân lớn nhất của số nhị phân 8 bit, 16 bit là bao nhiêu?

**6**. Đổi các số nhị phân 10110 và 10011011 sang số thập lục phân.

**7**. Đổi các số thập lục phân 1A, 7FF sang số nhị phân.

**8**. Sắp xếp các số theo thứ tự tăng dần: (1.1)2, (1.4)10, (1.5)16

**9**. Đổi giá trị biểu diễn

a) 548 sang hệ cơ số 5 b) 3124 sang hệ cơ số 7

**10**. Đổi các số nhị phân sau ra số trong hệ thập phân:

a) 001100 b) 011100 c) 101010 d)11100.011

e) 110011.10011 f) 1010101010.1

**11**.Đổi các số thập phân sau ra số trong hệ nhị phân:

a) 64 b) 100 c) 255

d) 34.75 e) 25.25 f) 27.1875

**12**.Đổi các số thập lục phân sau ra số trong hệ thập phân:

a) B52 b) ABCD c) D3.E

d) 1111.1 e) EBA.C

**13**.Đổi các số thập phân sau ra số trong hệ thập lục phân:

a) 2560 b) 6250 c) 16245

d) 204.125 e) 255.875 f) 631.25

**14**. Đổi các số thập lục phân sau ra số trong hệ nhị phân:

a) 568 b) A74 c) 1F.C

d) 239.4

**15**.Đổi các số nhị phân sau ra số trong hệ thập lục phân:

a) 1001.1111 b) 110101.011001 c) 101001111.111011

**CHƯƠNG 4. BỘ NHỚ CACHE**

**1**. Bộ vi xử lý 16 bit, kết nối bus (20 đường địa chỉ, 16 đường dữ liệu) với bộ nhớ tổ

chức dưới dạng các ngăn nhớ 8b. Hãy cho biết:

a. Kích thước word của BN trên

b. Dung lượng tối đa của bộ nhớ mà VXL có thể quản lý được.

**2**. Bộ nhớ chính: 2^16 byte, kích thước khối 8 byte, ánh xạ trực tiếp vào cache 32 đường.

a. 16 bit địa chỉ được chia thành các trường Tag, Line và Word như thế nào?

b. Các địa chỉ sau sẽ được lưu ở đường nào của cache?

0001 0001 0001 1011 1101 0000 0001 1101

1100 0011 0011 0100 1010 1010 1010 1010

c. Giả sử byte có địa chỉ 0001 1010 0001 1010 được lưu ở cache, các byte nào của

bộ nhớ chính cũng được lưu trên đường đó?

d. Có bao nhiêu byte có thể được lưu trên cache?

e. Khối nhớ thứ 36 được ánh xạ vào đường nào?

**3**. Bộ nhớ chính: 2^16 byte, kích thước khối 8 byte, ánh xạ trực tiếp vào cache 16 đường.

a. Kích thước ngăn nhớ 2 bytes, các trường Tag, Line và Word có kích thước bao

nhiêu?

b. Các địa chỉ sau sẽ được lưu ở đường nào của cache?

0001 0001 0001 1011 1101 0000 0001 1101

1100 0011 0011 0100 1010 1010 1010 1010

c. Có bao nhiêu byte có thể được lưu trên cache?

d. Khối nhớ thứ 19 được ánh xạ vào đường nào?

**4**. Xét VXL 32 bit, địa chỉ 32b. Bộ nhớ cache 16-Kbyte, ánh xạ tập kết hợp 4-way. Giả

sử một đường gồm 4 từ 32-bit (mỗi từ nhớ 32 bit, 1 đường có 4 từ). Xác định các

trường của địa chỉ được sử dụng để ánh xạ cache. Từ nhớ có địa chỉ ABCDE8F8 được

ánh xạ vào vị trí nào trong cache.

**5**. Bộ nhớ Cache 64 đường sử dụng ánh xạ tập kết hợp 4 đường. Bộ nhớ chính có 4K

khối, mỗi khối có kích thước 128 từ. Xác định định dạng địa chỉ bộ nhớ.

**CHƯƠNG 5: BỘ NHỚ TRONG**

**1**. Giả sử một word 8b cần được lưu trữ trong bộ nhớ là: 11000010. Sử dụng mã

Hamming SEC xác định các bit kiểm tra (check bits) được lưu trữ cùng từ trên. Viết

từ được lưu trữ.

**2**. Dữ liệu được lấy ra từ bộ nhớ: 000101001111. Xác định xem dữ liệu trên có bị lỗi hay

không. Nếu có thì sửa lỗi.

**3**. Dữ liệu được lấy ra từ bộ nhớ: 001101001110. Xác định xem dữ liệu trên có bị lỗi hay

không. Nếu có thì sửa lỗi.

**CHƯƠNG 6: BỘ NHỚ NGOÀI**

**1**. Một ổ đĩa cứng 256GB có 65.536 cylinder với 256 sector/track và 512 B/sector.

a. Tính số lượng đĩa và số đầu đọc mà ổ đĩa này cần (đĩa hai mặt, có đầu di chuyển).

b. Giả sử thời gian tìm kiếm trung bình là 11ms, thời gian trễ quay trung bình là 7ms,

tốc độ đọc là 100 MBps (thời gian truyền). Tính thời gian trung bình để đọc

400KB từ đĩa.

**2.** Một ổ đĩa từ có 8 mặt, mỗi mặt có 512 track, 64 sector trên mỗi track. Kích thước

sector là 1KB.

a. Dung lượng của ổ đĩa là bao nhiêu?

b. Thời gian tìm kiếm trung bình là 8 ms, thời gian di chuyển giữa các track là 1.5

ms, ổ quay với tốc độ 3600 rpm (vòng/phút). Tính thời gian truy cập trung bình

mỗi track. Tính thời gian cần thiết để đọc hoặc ghi một file dung lượng 5MB

**CHƯƠNG 9**

**1**. Biểu diễn các số thập phân sau sang dạng bù 2 8b và dấu-độ lớn 8b

a) - 54

b) 11

c) - 13

d) 145

**2.** Xác định miền giá trị của các dạng biểu diễn sau:

a. dấu-độ lớn 8b (tổng quát dấu-độ lớn n bit)

b. bù hai 8b (tổng quát bù hai n bit)

c. số nguyên không dấu 8b (tổng quát số nguyên ko dấu n bit)

**3.** Tính toán giá trị của các số biểu diễn dạng dấu- độ lớn 8b sau:

A = 0110 1011

B = 1101 1001

**4.** Tính toán giá trị của các số biểu diễn dạng mã bù 2 8b sau

A = 0110 1011

B = 1101 1001

**5.** Thực hiện phép toán sau biết các số biểu diễn dưới dạng bù 2 8b (chú ý kiểm tra tràn)

a. 1000 1001 + 0100 1000

b. 1111 0000 – 0111 0001

c. 0100 1111 + 0100 0000

**6.** Nêu nguyên tắc tràn của các phép toán với số nguyên bù 2

**7**. Biểu diễn các số thập phân sau sang dạng số thực dấu chấm động 32b

a. 10.125

b. -112.375

c. 25.5

**8**. Cho các số biểu diễn trong máy tính với định dạng số thực dấu chấm động 32b như

sau. Xác định giá trị thập phân tương ứng của số đó

a. 1100 0011 1110 1000 0000 0000 0000 0000

b. 0011 1110 1110 1100 0000 0000 0000 000