

## A. LÝ THUYẾT

### 1. Nêu quy tắc để chuyển đổi một số từ cơ số này sang cơ số khác

B1: Xác định cơ số ban đầu  $a$  (cơ số ban đầu) và cơ số muốn chuyển đổi sang  $b$  (cơ số đích)

B2: Chuyển đổi cơ số ban đầu về cơ số 10 (hệ thập phân) (nếu cần)

Giả sử số  $n = n_k \dots n_0$  ở cơ số  $a \Rightarrow N = \sum_{i=0}^k n_i a^i$  ở cơ số 10

B3: Chuyển từ cơ số 10 sang cơ số đích

\* Chuyển từ cơ số 10 sang cơ số 2:

- Đối với phần nguyên: Chia liên tục phần nguyên cho  $b$  cho đến khi kết quả bằng 0. Viết phần dư của từng phép tính theo chiều từ phải sang trái
- Đối với phần lẻ: Nhân liên tục phần lẻ với  $b$  cho đến khi kết quả bằng 1 (hoặc dừng khi đã đủ số chữ số sau dấu phẩy). Viết phần nguyên của từng phép tính theo chiều từ trái sang phải

\* Chuyển từ cơ số 10 sang cơ số 8:

Chuyển sang cơ số 2 trước  $\rightarrow$  cứ tổ hợp 3 chữ số trong cơ số 2 tương ứng với 1 chữ số trong cơ số 8. Từ dấu phẩy về 2 phía ta cứ nhóm 3 chữ số một trong cơ số 2 để chuyển tương ứng về cơ số 8. Nếu thiếu thì thêm chữ số 0 cho đủ bộ sao cho giá trị số đó không thay đổi

\* Chuyển từ cơ số 10 sang cơ số 16:

Làm tương tự như với cơ số 8 nhưng thay tổ hợp 3 chữ số bằng tổ hợp 4 chữ số

Ví dụ:

- $3F_{16} = ?_2 \Rightarrow 3F_{16} = 3 \cdot 16^1 + 15 \cdot 16^0 = 63_{10} = 11111_2$
- $25.06_{10} = 11001.000011(11)_2 = 31.03_8 = 19.0F_{16}$

### 2. Nêu nguyên tắc biểu diễn số nguyên có dấu

- Nguyên tắc chung: dùng  $n$  bit biểu diễn số nguyên có dấu  $A: a_{n-1} \dots a_0$

$\Rightarrow$  Giá trị của  $A = -a_{n-1} \cdot 2^{n-1} + (a_{n-2} \cdot 2^{n-2} + \dots + a_0 \cdot 2^0)$

Dải biểu diễn  $[-2^{n-1}, 2^{n-1} - 1]$  (VD:  $n = 8 \Rightarrow$  dải  $[-128, 127]$ )

- Có các cách biểu diễn phổ biến sau: dấu lượng, bù 1, bù 2 (được sử dụng phổ biến nhất trong máy tính)

- Biểu diễn bằng dấu lượng [1]:

- + Bit trái nhất (MSB) đánh dấu âm (1)/ dương (0)
- + Các bit còn lại (\*): biểu diễn giá trị tuyệt đối của số

- Biểu diễn bằng bù 1: Tương tự [1] nhưng nếu là số âm thì thực hiện phép đảo bit (0 thành 1, 1 thành 0) tất cả các bit của (\*)

- Biểu diễn bằng bù 2: Biểu diễn giống như số bù 1 rồi cộng thêm số 1 vào kết quả ở dạng nhị phân. Bù 2 khắc phục được 2 nhược điểm từ 2 phương pháp trên là:

- + Không đồng nhất trong cách biểu diễn, số 0 có 2 cách biểu diễn +0, -0
- + Bit nhớ phát sinh sau khi đã thực hiện phép tính phải được cộng tiếp vào kết quả, dễ gây nhầm lẫn

VD: tính bù 2 của -5 và 3

0000 0101 (dấu lượng của 5)

1111 1010 (bù 1 của -5, đổi bit vì đây là số âm)

1111 1011 (bù 2 của -5, cộng thêm bit 1 vào cuối của bù 1)

0000 0011 (dấu lượng của 3, bù 1, bù 2 của 3 vẫn giữ nguyên vì đây là số dương)

### 3. Khái niệm về thông tin, mã hóa và giải mã thông tin

- Thông tin: mô tả tất cả những gì đem lại sự hiểu biết, nhận thức cho con người có thể thu thập, lưu trữ và xử lý được, thông tin có thể ở dạng số (dấu phẩy động, tĩnh) hoặc phi số (logic, đa phương tiện, văn bản)

- Mã hoá thông tin: quá trình biến đổi dữ liệu thành một dạng dữ liệu khác có ý nghĩa khác với dữ liệu trước khi bị biến đổi ban đầu, chỉ cho phép một số người nhất định truy cập, mọi dữ liệu đều được mã hoá dưới dạng nhị phân (mã hoá cổ điển, 1 chiều, đối xứng và bất đối xứng)

- Giải mã thông tin: phương pháp để đưa từ dạng thông tin đã được mã hoá về dạng thông tin ban đầu, là quá trình ngược của quá trình mã hoá

### 4. Thực hiện các phép cộng 2 số nhị phân có dấu số 8 bit

B1: Biểu diễn 2 số dưới dạng bù 2

B2: Thực hiện phép cộng bit nhị phân. Nếu có nhớ từ bit cao nhất (bit số 9) thì bỏ qua nhớ này. Kết quả cuối cùng là biểu diễn bù 2 của số thập phân kết quả

VD:  $-5 + 3 \Rightarrow$  cộng bù 2 của 2 số trên

1111 1011  
+ 0000 0011  
= 1111 1110 (đây chính là bù 2 của số -2)

### 5. Trình bày về biểu diễn số thực dấu phẩy tĩnh, lấy ví dụ với số 16 bit

- Biểu diễn số thực dấu phẩy tĩnh là cố định số bit được dành riêng để biểu diễn phần nguyên và phần thập phân của một số. Ví dụ nếu có 16 bit thì có thể chia thành 8 bit cho phần nguyên và 8 bit cho phần thập phân.

- Chú ý: với số âm, bit dấu thường được thêm vào đầu (0 cho dương, 1 cho âm)  $\Rightarrow$  phải điều chỉnh số lượng bit dùng cho phần nguyên và phần thập phân tương ứng

VD: Biểu diễn số 10.625

- Đổi phần nguyên (10) sang nhị phân: 00001010

- Đổi phần thập phân (0.625) sang nhị phân: .101

$\Rightarrow$  Biểu diễn dấu phẩy tĩnh 16 bit: 00001010.10100000 (các bit 0 cuối thêm cho đủ 16 bit)

### 6. Trình bày về biểu diễn số thực dấu phẩy động 32 bit, ví dụ

- Biểu diễn số thực dấu phẩy động (tiêu chuẩn IEEE 754) là cách biểu diễn số thực trong máy tính phổ biến nhất.

B1: Chuyển đổi số về dạng nhị phân rồi chuẩn hóa theo công thức:  $(-1)^S * (1.M) * 2^{E-127}$

B2: Lưu dấu S. Tính và lưu số mũ E trong hệ nhị phân. Lưu phần nhị phân M

B3: Biểu diễn dấu phẩy động 32 bit: S E M (S = 1 bit dấu với 0 cho dương, 1 cho âm; E = 8 bit, M = 23 bit và thêm 0 ở cuối nếu cần)

VD: Biểu diễn số 10.625

- Đổi số sang nhị phân:  $1010.101 \Rightarrow$  chuẩn hóa  $= 1.010101 * 2^3$
- S = 0 do số dương,  $E = 3 + 127 = 130_{10} = 10000010_2$ , M = 010101

$\Rightarrow$  Biểu diễn dấu phẩy động 32 bit: 0100 0001 0010 1010 0000 0000 0000 0000 (các bit 0 cuối thêm cho đủ 32 bit)

## 7. Trình bày các thành phần cơ bản của máy tính

Thành phần cơ bản của một máy tính bao gồm:

- **Bộ xử lý trung tâm (CPU: Central Processing Unit):** điều khiển hoạt động của máy tính và xử lý dữ liệu

CPU lấy lệnh từ bộ nhớ trong và lấy các số liệu mà lệnh đó xử lý. Bộ xử lý trung tâm gồm có hai phần: phần thi hành lệnh và phần điều khiển. Phần thi hành lệnh bao gồm bộ làm toán và luận lý (ALU: Arithmetic And Logic Unit) và các thanh ghi. Nó có nhiệm vụ làm các phép toán trên số liệu. Phần điều khiển có nhiệm vụ đảm bảo thi hành các lệnh một cách tuần tự và tác động các mạch chức năng để thi hành các lệnh.

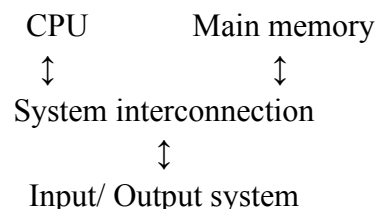
- **Bộ nhớ chính (Main memory):** chứa chương trình và dữ liệu đang được sử dụng. Có 2 đơn vị bộ nhớ chính trong máy tính

- + RAM: Random Access Memory - là bộ nhớ truy cập ngẫu nhiên chứa các lệnh hay số liệu. Mỗi ô nhớ đều có một địa chỉ cho phép CPU truy cập nhanh chóng với thời gian thâm nhập vào một ô nhớ bất kỳ là như nhau. Dung lượng một ô nhớ thông thường là 8 bit (1 Byte).
- + ROM: Read Only Memory - là bộ nhớ chỉ đọc chứa các hướng dẫn đã được cài đặt sẵn trong máy tính, nó chứa thông tin cần thiết để khởi động máy tính

- **Hệ thống vào ra (Input/ Output system):** trao đổi thông tin giữa máy tính với bên ngoài

Các hệ thống vào ra phổ biến là: bộ lưu trữ ngoài, màn hình, máy in, bàn phím, chuột, máy quét ảnh, các giao diện mạng cục bộ hay mạng diện rộng... Bộ tạo thích ứng là một vi mạch tổng hợp (chipset) kết nối giữa các hệ thống bus có các tốc độ dữ liệu khác nhau.

- **Liên kết hệ thống (System interconnection):** kết nối và vận chuyển thông tin giữa các thành phần với nhau



## 8. Trình bày về bộ nhớ từ, so sánh HDD, FDD, băng từ

- Bộ nhớ từ là bộ nhớ điện tĩnh, sử dụng đặc tính từ tính của vật liệu để lưu trữ thông tin dưới dạng trạng thái từ tính biểu diễn các bit 0 với 1. Bộ nhớ này thường có một số nhược điểm sau: tuổi thọ ngắn, tỏa nhiều nhiệt lượng, có tiếng ồn khi hoạt động, tốc độ truy xuất dữ liệu bị phụ thuộc vào giới hạn của tốc độ quay. Các loại bộ nhớ từ khác nhau được phân loại dựa trên cách chúng sử dụng các đặc tính này.

	<b>Ổ đĩa cứng HDD (Hard disk drive)</b>	<b>Ổ đĩa mềm FDD (Floppy disk drive)</b>	<b>Băng từ (tape)</b>
Cơ chế hoạt động	Dùng các đĩa từ cứng quay và đầu đọc/ghi di động để lưu trữ và truy xuất dữ liệu	Dùng đĩa mềm có thể tháo rời và đầu đọc/ghi di động để lưu trữ và truy xuất dữ liệu	Dùng băng từ dài để lưu trữ dữ liệu, đầu đọc/ghi di động để truy xuất dữ liệu
Dung lượng lưu trữ	Lớn, từ vài GB đến vài TB	Rất thấp, thường chỉ khoảng 1.44 MB	Rất lớn, có thể lên đến PB
Tốc độ đọc/ ghi	Trung bình đến nhanh, tùy thuộc vào mô hình	Chậm	Chậm, nhưng có thể được tăng cường với kỹ thuật "streaming"
Truy cập ngẫu nhiên	Có, có thể truy cập dữ liệu ở bất kỳ vị trí nào trên đĩa	Có, nhưng tốc độ truy cập ngẫu nhiên thường chậm	Không, phải tua băng để truy cập dữ liệu
Độ tin cậy	Trung bình, có thể hỏng nếu di chuyển hoặc rung lắc	Thấp, đĩa mềm dễ bị hỏng và dữ liệu dễ bị mất	Cao, băng từ thường bền và dữ liệu được lưu trữ an toàn
Phù hợp cho	Lưu trữ dữ liệu hàng ngày, hệ thống chạy hệ điều hành và ứng dụng	Hiện nay gần như không còn sử dụng. Trước đây được sử dụng để chuyển dữ liệu nhỏ giữa các máy tính	Sao lưu dữ liệu lớn và lưu trữ dài hạn
Chi phí	Trung bình, tùy thuộc vào dung lượng và mô hình	Thấp, nhưng hiện nay khó tìm để mua mới	Thấp cho mỗi GB lưu trữ, nhưng thiết bị đọc/ghi băng từ thường khá đắt

## 9. Trình bày về bộ nhớ quang, so sánh CD-DVD-BlueRay

- Bộ nhớ quang là một dạng thiết bị lưu trữ dữ liệu không tự động, nghĩa là dữ liệu lưu trữ không bị mất khi tắt nguồn điện. Bộ nhớ quang sử dụng ánh sáng laser để đọc và (có thể) ghi dữ liệu.

- Cấu trúc của một đĩa quang thường bao gồm một lớp nhựa polycarbonate trong suốt, một lớp phản xạ kim loại, và một lớp bảo vệ nhựa hoặc lacquer. Dữ liệu được lưu trữ dưới dạng một chuỗi các rãnh và điểm đại diện cho dữ liệu 1 và 0, được ghi vào lớp polycarbonate. Khi đĩa được quay, một tia laser chiếu vào lớp polycarbonate và phản xạ trở lại một cảm biến. Sự khác biệt giữa pit và land dẫn đến sự thay đổi trong tín hiệu phản xạ, cho phép cảm biến đọc dữ liệu.

	<b>CD (Compact Disc)</b>	<b>DVD (Digital Versatile Disc)</b>	<b>BlueRay Disc</b>
Cơ chế hoạt động	Sử dụng tia laser hồng ngoại	Sử dụng tia laser đỏ	Sử dụng tia laser tím

Dung lượng lưu trữ	Khoảng 700 MB	4.7 GB (đơn lớp), 8.5 GB (đôi lớp)	25 GB (đơn lớp), 50 GB (đôi lớp), 100/128 GB (BDXL)
Tốc độ đọc/ ghi	1x - 52x	1x - 24x	1x - 12x
Độ phân giải video	Không phù hợp để lưu trữ video với chất lượng cao	Phù hợp với video độ phân giải SD và HD	Phù hợp với video độ phân giải HD, Full HD, 3D và 4K
Độ tin cậy	Cao	Cao	Cao
Phù hợp cho	Lưu trữ âm nhạc, dữ liệu nhỏ	Lưu trữ video, dữ liệu lớn hơn	Lưu trữ video HD, 4K, dữ liệu lớn
Chi phí	Thấp	Trung bình	Cao so với CD, DVD

## 10. Trình bày về bộ nhớ ROM, RAM (so sánh SRAM, DRAM)

- ROM:

- + Bộ nhớ chỉ đọc (thông tin chỉ được đọc mà không cho phép ghi vào) và chỉ cho phép ghi dữ liệu 1 lần
- + Dùng để lưu trữ các thông số kỹ thuật cố định như firmware , BIOS, các thông số cấu hình và hướng dẫn chương trình khởi động máy tính được nạp bởi nhà sản xuất
- + ROM có tính chất bền vững và an toàn, thông tin lưu trữ không bị mất khi tắt máy

- RAM:

- + Bộ nhớ truy cập ngẫu nhiên
- + Được sử dụng cho bộ nhớ chính trong máy tính để lưu trữ các thông tin thay đổi, và các thông tin được sử dụng hiện hành. Dữ liệu trong RAM có thể được đọc và ghi nhanh chóng, giúp máy tính xử lý hiệu quả hơn so với việc đọc trực tiếp trên đĩa cứng
- + RAM là nơi mà chương trình và dữ liệu được tải vào bộ nhớ trước khi được lưu trữ. Thông tin lưu trữ trên RAM là tạm thời nghĩa là khi tắt máy sẽ bị mất

	<b>SRAM (Static RAM)</b>	<b>DRAM (Dynamic RAM)</b>
Cấu trúc	Thiết kế SRAM khá phức tạp do sử dụng 6 bóng bán dẫn cho một khối bộ nhớ	DRAM chỉ cần dùng 1 bóng bán dẫn và 1 tụ điện cho một khối bộ nhớ
Chức năng	Ứng dụng trong bộ nhớ Cache	Ứng dụng trong bộ nhớ chính
Tốc độ	SRAM nhanh hơn do là bộ nhớ trên chip có thời gian truy cập nhỏ	DRAM là bộ nhớ ngoài chip có thời gian truy cập lớn, phải liên tục refresh dữ liệu để duy trì năng lượng trong các tụ điện
Dung lượng	Dung lượng lưu trữ của SRAM là	Dung lượng lưu trữ của DRAM là 1

	1MB đến 16MB	GB đến 16 GB
Tiêu thụ năng lượng	SRAM cần nguồn điện liên tục, tiêu thụ năng lượng nhiều hơn so với DRAM do yêu cầu một lượng transistor lớn để lưu trữ dữ liệu	DRAM tiêu thụ năng lượng ít hơn so với SRAM do có cấu trúc đơn giản hơn và ít transistor hơn
Chi phí	Đắt hơn	Rẻ hơn

### 11. Trình bày về kỹ thuật ống dẫn, nêu các xung đột và phương pháp giải quyết

- Kỹ thuật ống dẫn (pipeline) là một cách để tăng cường hiệu suất của CPU bằng cách cho phép nhiều lệnh được thực hiện đồng thời (chia quá trình thực hiện lệnh thành nhiều bước có thể thực hiện cùng lúc)
- Ống dẫn hoạt động như một dây chuyền sản xuất, trong đó mỗi lệnh đi qua các giai đoạn khác nhau của quá trình thực hiện: fetch (lấy), decode (giải mã), execute (thực hiện), memory access (truy cập bộ nhớ), và write back (ghi lại). Trong một thời điểm, mỗi giai đoạn của ống dẫn đều đang thực hiện một phần của một lệnh khác nhau.
- Một số xung đột có thể gặp phải và hướng giải quyết:
  - + Xung đột dữ liệu (data hazard): Khi một lệnh phụ thuộc vào kết quả của một lệnh trước đó mà vẫn đang trong ống dẫn  $\Rightarrow$  sử dụng các kỹ thuật như forwarding (chuyển tiếp) hoặc stalling (dừng)
  - + Xung đột điều khiển (control hazard): Khi có một lệnh nhảy (như lệnh nhảy có điều kiện) và lệnh tiếp theo để thực hiện không rõ ràng cho đến khi lệnh nhảy được hoàn thành  $\Rightarrow$  sử dụng các kỹ thuật như branch prediction (dự đoán nhánh), delay slot (khe trễ), hoặc branch target buffer
  - + Xung đột cấu trúc (structural hazard): Khi một phần của bộ xử lý (bộ nhớ hoặc ALU) được yêu cầu để thực hiện nhiều công việc cùng một lúc  $\Rightarrow$  tăng cường phần cứng

### 12. Trình bày về máy tính song song

- Máy tính song song là một loại hệ thống máy tính mà trong đó nhiều bộ vi xử lý (CPU) làm việc cùng một lúc để thực hiện các tác vụ. Hệ thống song song có thể xử lý nhiều phần của chương trình hoặc nhiều chương trình cùng một lúc  $\Rightarrow$  rất hữu ích để thực hiện một phép tính phức tạp, giúp tiết kiệm thời gian.
- Có 2 loại cơ bản của máy tính song song:
  - + Máy tính song song dữ liệu (Data Parallelism): cùng một chương trình được thực hiện trên các phần khác nhau của dữ liệu  $\Rightarrow$  giải quyết các vấn đề cần xử lý một lượng lớn dữ liệu (VD: xử lý hình ảnh, xử lý tín hiệu, và mô phỏng vật lý)
  - + Máy tính song song tác vụ (Task Parallelism): các tác vụ khác nhau của cùng một chương trình được thực hiện đồng thời  $\Rightarrow$  giải quyết các vấn đề cần nhiều tác vụ được thực hiện cùng lúc (VD: hệ thống học máy, AI, và hệ thống cơ sở dữ liệu lớn)
- Máy tính song song cần các thuật toán và phần mềm phức tạp hơn so với máy tính tuần tự, nhưng nó cung cấp khả năng xử lý mạnh mẽ và hiệu quả hơn đáng kể  $\Rightarrow$  là cơ sở cho hầu hết các siêu máy tính và hệ thống đám mây hiện đại.

### 13. Trình bày về cảm biến hình ảnh

- Cảm biến hình ảnh, thường gặp trong máy ảnh kỹ thuật số và máy quay video, là một thiết bị chuyển đổi hình ảnh từ thế giới thực sang dạng điện tử.
- Nguyên tắc hoạt động:
  - + Nhận ánh sáng: Cảm biến hình ảnh bao gồm hàng triệu pixel nhỏ. Mỗi pixel chứa một photodiode, một thiết bị chuyển đổi ánh sáng thành dòng điện
  - + Chuyển đổi ánh sáng thành dòng điện: Khi ánh sáng rơi vào photodiode, nó tạo ra các hạt electron. Số lượng electron tạo ra phụ thuộc vào lượng ánh sáng rơi vào diode: càng nhiều ánh sáng, càng nhiều electron
  - + Chuyển đổi dòng điện thành dữ liệu số: Dòng điện từ mỗi pixel sau đó được chuyển đổi thành dữ liệu số bằng bộ chuyển đổi tương tự-số (ADC). Dữ liệu số này sau đó được xử lý để tạo thành hình ảnh kỹ thuật số
- Hai loại cảm biến hình ảnh phổ biến nhất là CMOS (Complementary Metal Oxide Semiconductor) và CCD (Charge Coupled Device). CCD tạo ra hình ảnh chất lượng cao và nhiễu thấp hơn, nhưng chúng tiêu thụ nhiều năng lượng hơn và thường đắt hơn so với cảm biến CMOS và ngược lại

### 14. Trình bày về chuẩn giao tiếp I2C

- I2C (Inter-Integrated Circuit) là một giao thức giao tiếp nối tiếp đồng bộ. được phát triển dùng để truyền dữ liệu giữa một bộ xử lý trung tâm với nhiều IC trên cùng một board mạch chỉ sử dụng hai đường truyền tín hiệu
- Cấu tạo: I2C sử dụng 2 đường truyền tín hiệu
  - + SCL - Serial Clock Line: Tạo xung nhịp đồng hồ do Master phát đi
  - + SDA - Serial Data Line: Đường truyền nhận dữ liệu.
- Cách thức hoạt động: Giao tiếp I2C bao gồm quá trình truyền nhận dữ liệu giữa các thiết bị chủ tớ (Master - 1 vi điều khiển có nhiệm vụ điều khiển đường SCL và gửi nhận dữ liệu hay lệnh thông qua đường SDA đến các Slave - các thiết bị, thường là IC, hoặc có thể là vi điều khiển, mà nhận dữ liệu lệnh và tín hiệu từ Master)
  - + Master gửi điều kiện khởi động đến mọi slave được kết nối bằng cách chuyển đường SDA từ mức điện áp cao sang mức điện áp thấp trước khi chuyển đường SCL từ mức cao xuống mức thấp.
  - + Master gửi cho mỗi slave địa chỉ 7 hoặc 10 bit của slave mà nó muốn giao tiếp, cùng với bit đọc/ghi.
  - + Mỗi slave sẽ so sánh địa chỉ được gửi từ master với địa chỉ của chính nó. Nếu địa chỉ trùng khớp, slave sẽ trả về một bit ACK bằng cách kéo dòng SDA xuống thấp cho một bit. Nếu địa chỉ từ master không khớp với địa chỉ của slave, slave rời khỏi đường SDA cao.
  - + Master gửi hoặc nhận khung dữ liệu.
  - + Sau khi mỗi khung dữ liệu được chuyển, thiết bị nhận trả về một bit ACK khác cho thiết bị gửi để xác nhận đã nhận thành công khung.
  - + Để dừng truyền dữ liệu, master gửi điều kiện dừng đến slave bằng cách chuyển đổi mức cao SCL trước khi chuyển mức cao SDA
- Ưu điểm: chỉ dùng 2 dây, phần cứng ít phức tạp so với UART, một thiết bị truyền có thể có nhiều thiết bị nhận (master - slaves)  $\Rightarrow$  do tính đơn giản của nó nên loại giao thức này được



sử dụng rộng rãi cho giao tiếp giữa vi điều khiển và mảng cảm biến, các thiết bị hiển thị, thiết bị IoT, ...

- Nhược điểm: chậm hơn SPI, kích thước khung dữ liệu giới hạn 8bit

### **15. Trình bày về chuẩn giao tiếp 1-wire**

- 1-Wire là một chuẩn giao tiếp hệ thống bus (không đồng bộ và bán song công) chỉ sử dụng một dây để truyền dữ liệu và năng lượng giữa các thiết bị điện tử. Dây này thường được kết nối với nguồn điện thông qua một điện trở kéo lên (pull-up resistor)  $\Rightarrow$  tiết kiệm không gian và giảm chi phí hệ thống

- Tuân theo mô hình master - slave, mỗi một thiết bị slave sẽ có 64 bit địa chỉ lưu trữ duy nhất trong ROM của mình  $\Rightarrow$  cho phép nhiều thiết bị có thể kết nối với cùng một dây mà không gây ra xung đột

- Chuẩn giao tiếp 1-Wire sử dụng khái niệm time slot (khe thời gian), khoảng thời gian trong đó mức logic 1 hoặc 0 sẽ được ghi hoặc đọc. Chế độ Standard, time slot = 60 $\mu$ s, chế độ Override, time slot = 8 $\mu$ s.

- Có 3 thao tác: Reset, Read, Write. Các bước truyền dữ liệu:

- + Write 1 (Truyền đi bit 1): Master kéo xuống 0 một khoảng A( $\mu$ s) rồi về mức 1 một khoảng B( $\mu$ s)
- + Write 0 (Truyền đi bit 0): Master kéo xuống 0 trong khoảng C( $\mu$ s) rồi trả về 1 trong khoảng D( $\mu$ s)
- + Read (Đọc một bit): Master kéo xuống 0 khoảng A( $\mu$ s) rồi trả về 1, delay khoảng E( $\mu$ s) rồi đọc giá trị slave gửi về, delay F( $\mu$ s)
- + Reset (Chuẩn bị giao tiếp): Master kéo xuống 0 một khoảng H ( $\mu$ s) rồi nhả lên mức 1 sau đó delay I ( $\mu$ s) rồi đọc giá trị slave trả về và delay J ( $\mu$ s). Nếu giá trị này bằng 0 thì cho phép giao tiếp, nếu bằng 1 đường truyền lỗi hoặc slave đang bận

## **B. THỰC HÀNH**

1. Viết chương trình nhấp nháy đèn LED theo chu kỳ 1s; Bấm BT1 đèn đều tắt (sử dụng đa luồng được điểm tối đa)

2. Viết chương trình bấm BT1 hiển thị hello – world ra terminal, ra màn hình LCD, bấm BT2 lần 1 từng ký tự trong dòng chữ đó chạy từ phải qua trái, BT2 lần 2 thì từng chữ chạy ngược lại từ trái qua phải

3. Viết chương trình bấm BT1 lần 1 LED sáng, bấm BT1 lần 2 LED tắt, lặp đi lặp lại

4. Viết chương trình hiển thị \*hello world trên màn hình LED matrix, từng ký tự một chạy từ trái sang phải

5. Viết chương trình đọc giá trị cảm biến siêu âm, hiển thị liên tục giá trị khoảng cách lên LCD, nếu khoảng cách nhỏ hơn 6 thì bật role. (viết chương trình mô phỏng cảm biến lùi của ô tô – phát triển các chức năng được điểm tối đa)



6. Viết chương trình đếm ngược từ 0-9 hiển thị trên led matrix, trong quá trình đếm nếu bấm các phím lần lượt 3-2-4 thì sẽ dừng đếm, nếu bấm sai thứ tự trên quá trình đếm tiếp tục, khi số đếm bằng 0 sẽ hiện lên chữ hello-world trên màn hình và bật rơ le 1 + quay động cơ DC
7. Viết chương trình mô phỏng bật tắt đèn đường sử dụng cảm biến ánh sáng, hiển thị trời sáng – tối và trạng thái rơ le lên màn hình LCD
8. Viết chương trình nhận các phím bấm trên điều khiển từ xa và hiện các phím chức năng lên màn hình LCD
9. Viết chương trình bấm BT1, chụp ảnh và khoanh vùng những vật có màu đỏ, hiện ảnh lên màn hình
10. Viết chương trình đọc giá trị nhiệt độ, độ ẩm của cảm biến DHT11 hiển thị giá trị lên LCD
11. Viết chương trình mỗi lần bấm BT1, động cơ servo quay một góc 5 độ, khi quay đến 180 độ thì quay về 0, hiển thị góc lên LCD
12. Viết chương trình mỗi lần bấm BT1, động cơ DC tăng tốc độ lên 10%, hiển thị tốc độ lên màn hình LCD
13. Viết chương trình hiển thị các thanh trượt điều chỉnh cường độ trong không gian màu HSV để điều chỉnh ảnh từ camera