

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**  
**KHOA KỸ THUẬT ĐIỆN TỬ 1**



**BÁO CÁO BÀI TẬP LỚN**  
**KIẾN TRÚC MÁY TÍNH**

**Đề bài:**

**Mô phỏng giao tiếp LCD 16x2**

**Nhóm: 04**

**Giảng viên hướng dẫn : Đình Quang Ngọc**  
**Sinh viên thực hiện : Nguyễn Danh Dương – B22DCDT071**  
**Trịnh Quốc Việt – B22DCDT330**  
**Ngô Nhật Minh – B22DCDT191**  
**Ngô Xuân Điện – B22DCDT083**

**Hà Nội, 2024**

# MỤC LỤC

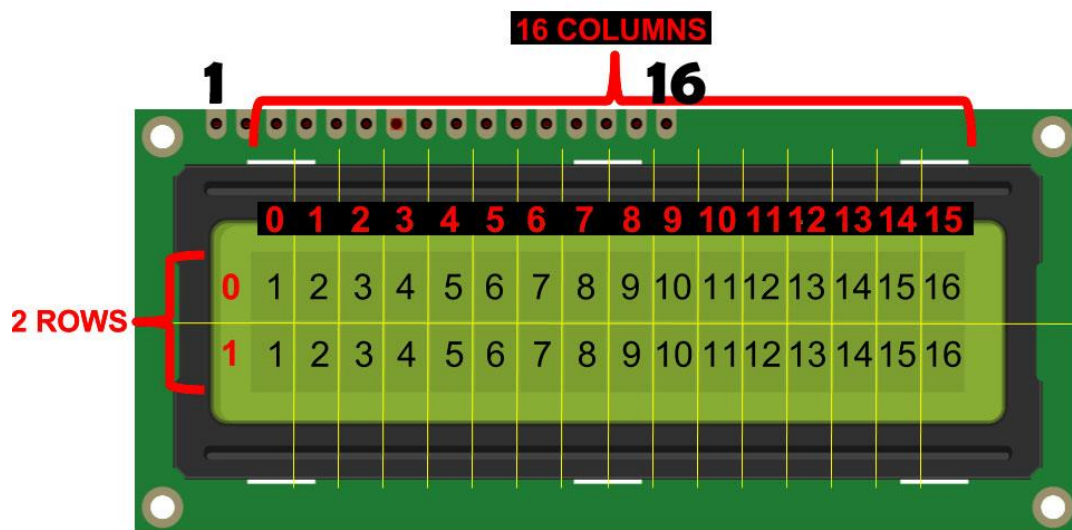
CHƯƠNG 1. GIỚI THIỆU VỀ ĐỀ TÀI .....	2
1.1. Giới thiệu về LCD 16x2 .....	2
1.1.1. Sơ đồ chân và các câu lệnh cần thiết .....	2
1.1.2. Cách biểu diễn kí tự trên LCD 16x2 .....	5
1.2. Giới thiệu mạch mô phỏng .....	5
1.2.1. CPU 8086 .....	5
1.2.2. IC 74HC373 .....	6
1.2.3. IC 8255A .....	7
CHƯƠNG 2. NỘI DUNG CHÍNH .....	10
2.1. Giải thích code .....	10
CHƯƠNG 3. KẾT QUẢ .....	13
3.1. Kết quả mô phỏng .....	13
3.2. Trả lời câu hỏi bổ sung .....	13

# Chương 1. GIỚI THIỆU VỀ ĐỀ TÀI

## 1.1. Giới thiệu về LCD 16x2

### 1.1.1. Sơ đồ chân và các câu lệnh cần thiết

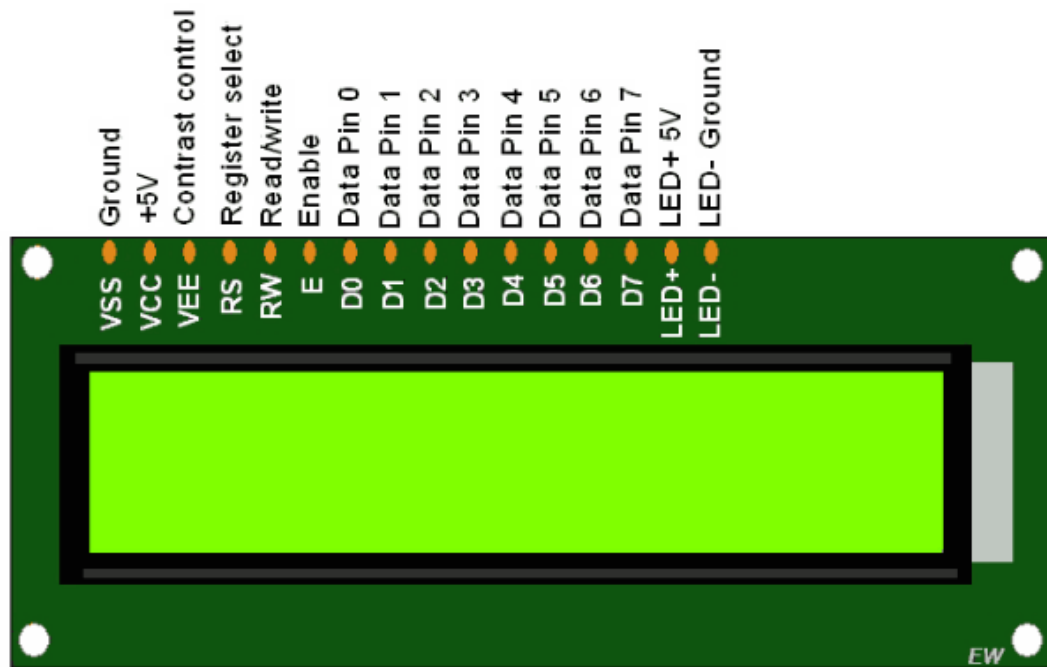
LCD 16x2 là màn hình hiển thị ký tự phổ biến, có khả năng hiển thị 32 ký tự trên 2 dòng. Sử dụng công nghệ tinh thể lỏng và bộ điều khiển HD44780, nó hỗ trợ giao tiếp song song 4-bit hoặc 8-bit. Với khả năng hiển thị rõ ràng, dễ sử dụng và giá thành hợp lý, LCD 16x2 được ứng dụng rộng rãi trong các hệ thống nhúng và thiết bị điện tử.



Màn hình LCD 16x2 gồm 16 hàng và 2 cột, tổng cộng có 32 ký tự. Mỗi ký tự được hiển thị trên màn hình bao gồm 5x8 điểm ảnh (pixel).



Sơ đồ chân:



Chức năng:

VSS: chân nối đất cho LCD được nối với GND của mạch điều khiển.

VDD: chân cấp nguồn cho LCD.

VEE: điều chỉnh độ tương phản của LCD.

RS: chân chọn thanh ghi lệnh (mức logic 0) hoặc thanh ghi dữ liệu (mức logic 1).

RW: chân chọn chế độ ghi (mức logic 0) hoặc chế độ đọc (mức logic 1).

E: chân cho phép, các lệnh chỉ được cho phép khi có xung của chân này.

D0-D7: các chân truyền dữ liệu.

LED+ : chân nguồn đèn trên LCD.

LED-: chân GND của đèn trên LCD.

Các câu lệnh quan trọng:

Mã Hex	Lệnh cho LCD
01	Xóa màn hình
02	Quay trở lại
04	Giảm con trỏ (chuyển con trỏ sang trái)
06	Tăng con trỏ (chuyển con trỏ sang phải)
05	Chuyển màn hình sang phải
07	Chuyển màn hình sang trái
08	Tắt hiển thị, tắt con trỏ
0A	Tắt hiển thị, bật con trỏ
0C	Bật hiển thị, con trỏ tắt
0E	Bật hiển thị, con trỏ nhấp nháy
10	Chuyển vị trí con trỏ sang bên trái
14	Chuyển vị trí con trỏ sang bên phải
18	Chuyển toàn bộ màn hình sang bên trái
1C	Chuyển toàn bộ màn hình sang bên phải
80	Buộc con trỏ đến đầu (dòng thứ 1)
C0	Buộc con trỏ đến đầu (dòng thứ 2)
38	2 hàng và ma trận 5×8

### 1.1.2. Cách biểu diễn kí tự trên LCD 16x2

Trong màn LCD, mỗi kí tự được biểu diễn trong ma trận 5x8 trong đó 5 là số cột và 8 là số hàng.

CG-RAM ADDRESS	Byte Address	C1	C2	C3	C4	C5	Pattern for b is
Starting Address of CG-RAM.	0 0 0 0	1	0	0	0	0	Row1 0x10=00010000
	0 0 0 1	1	0	0	0	0	Row2 0x10=00010000
	0 0 1 0	1	0	1	1	0	Row3 0x16=00010110
	0 0 1 1	1	1	0	0	1	Row4 0x19=00011001
	0 1 0 0	1	0	0	0	1	Row5 0x11=00010001
	0 1 0 1	1	0	0	0	1	Row6 0x11=00010001
	0 1 1 0	1	1	1	1	0	Row7 0x1E=00011110
	0 1 1 1	0	0	0	0	0	Cursor position

Diagram showing CG-RAM ADDRESS: 0 1 0 0 0 0 0. Hexadecimal=0x40, Decimal=64.

Annotations: First three bits are treated as Dont cares. \* Rest five bits are loaded at the address. electronicsforu.com

Để hiển thị kí tự b, ta xác định các bit cần bật trong ma trận, với mỗi hàng ta sẽ có các mã hex tương ứng, riêng hàng số 8 sẽ được giữ lại để hiển thị con trỏ nếu cần.

Mảng để tạo chữ cái 'b' là: { 0x10, 0x10, 0x16, 0x19, 0x11, 0x11, 0x1E }.

## 1.2. Giới thiệu mạch mô phỏng

### 1.2.1. CPU 8086

RESET: Tín hiệu khởi động lại CPU 8086. Khi RESET = 1 kéo dài ít nhất trong thời gian 4 chu kì đồng hồ thì 8086 bị buộc phải khởi động lại.

MN/MX (MIN/ MAX):

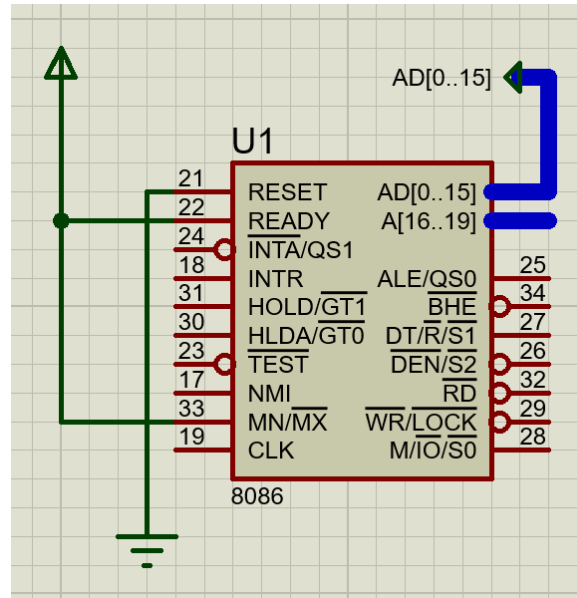
- Tín hiệu ở chân này cho biết bộ xử lí sẽ hoạt động ở chế độ nào.
- Chân MN/MX nối thẳng vào nguồn điện +5V mà không qua điện trở.

ALE (Address Latch Enable): Xung cho phép lưu trữ địa chỉ.

- ALE = 1 : trên bus dồn kênh AD có các địa chỉ của thiết bị vào/ra.
- ALE = 0: khi CPU bị treo.

AD0 - AD15 [I; O: tín hiệu vào và ra]: Bus địa chỉ/dữ liệu. Xung ALE (Address Latch Enable) sẽ báo cho mạch ngoài biết khi nào trên đường đó có tín hiệu dữ liệu (ALE = 0) hoặc địa chỉ (ALE = 1).

=> Các tín hiệu AD0 – AD15 được nối vào cùng 1 bus



### 1.2.2. IC 74HC373

LE(Latch Enable) : Đầu vào kích hoạt lưu trữ (hoạt động khi ở mức cao).

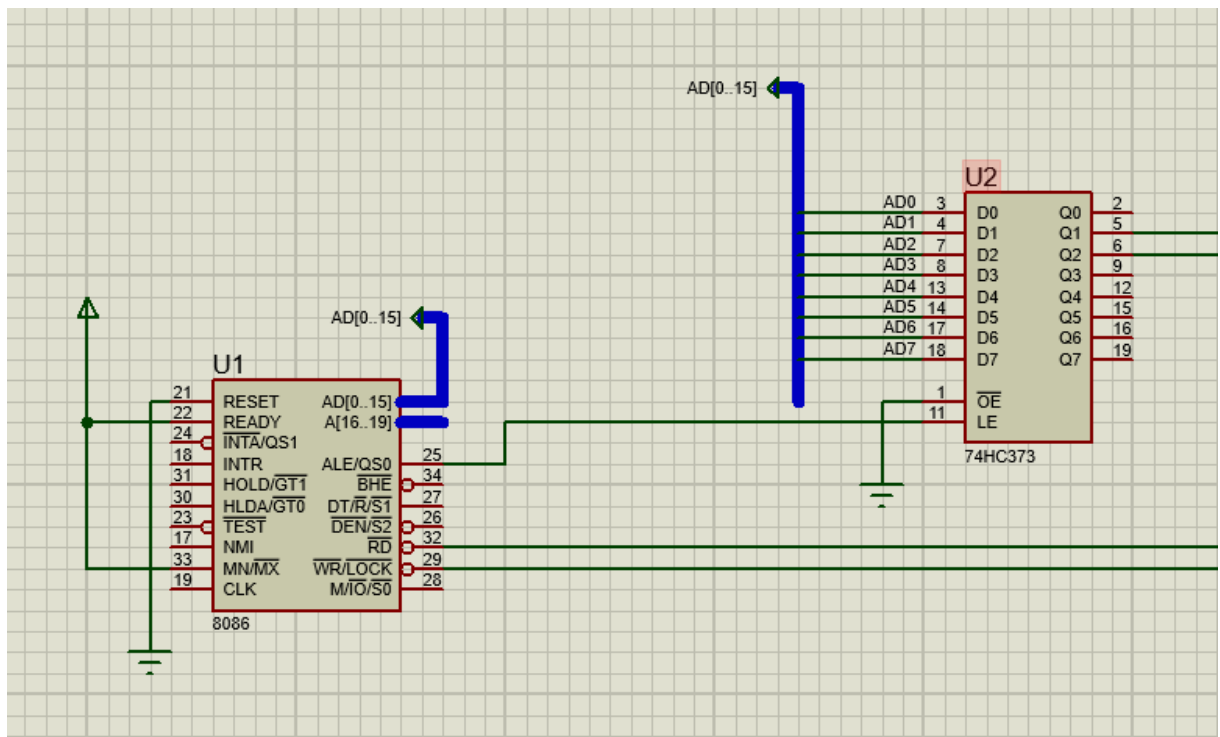
- LE ở mức cao (LE = 1): Dữ liệu ở đầu vào sẽ được lưu trữ. Đầu ra thay đổi mỗi khi mỗi đầu vào tương ứng thay đổi.
- LE ở mức thấp (LE = 0): lưu trữ thông tin ở các đầu vào tại thời điểm thiết lập trước khi quá trình chuyển đổi từ cao sang thấp của LE (LE xuống thấp thì các đầu ra  $Q_n$  sẽ được giữ lại đến khi LE trở lại mức cao).

Q0 – Q7 : 3 trạng thái đầu ra.

OE (Out Enable) : Kích hoạt 3 trạng thái đầu ra.

ALE (Address Latch Enable) của 8086 được kết nối với LE của 74HC373.

Dữ liệu đầu vào của 74CH373 từ D0 – D7 là các tín hiệu đầu vào cùng nối chung 1 bus.



### 1.2.3. IC 8255A

RESET: để khởi động 8255A khi cấp điện, khi bị reset các thanh ghi bên trong 8255A đều bị xóa và 8255A ở trạng thái sẵn sàng làm việc.

CS (Chip Select) đầu vào ở mức thấp ( $CS = 0$ ) cho phép kết nối giữa 8255A với CPU 8086.

=> CS(Chip Select) và RESET cần ở mức thấp.

=>  $CS(Chip\ Select) = 0$

$RESET = 0$

RD (READ): đầu vào ở mức thấp ( $RD = 0$ ) cho phép 8255A được đưa dữ liệu hoặc thông tin trạng thái tới CPU trên bus dữ liệu.

WR (WRITE): đầu vào ở mức thấp ( $WR = 0$ ) cho phép CPU ghi dữ liệu hoặc điều khiển các từ vào 8255A.

RD và WR/LOCK của 8086 được nối với của 8255A.

Các tín hiệu D0 – D7 của 8255A được nối vào cùng 1 bus.



A0, A1: Cổng chọn 0 và cổng chọn 1, địa chỉ các cổng (Port). Các tín hiệu đầu vào này kết hợp với đầu vào RD và WR, điều khiển việc lựa chọn một trong ba cổng (Port) hoặc điều khiển thanh ghi.

Đầu ra Q1, Q2 của 74HC373 được nối lần lượt trực tiếp với A0, A1 của 8255A để chọn chế độ hoạt động của IC 8255A.

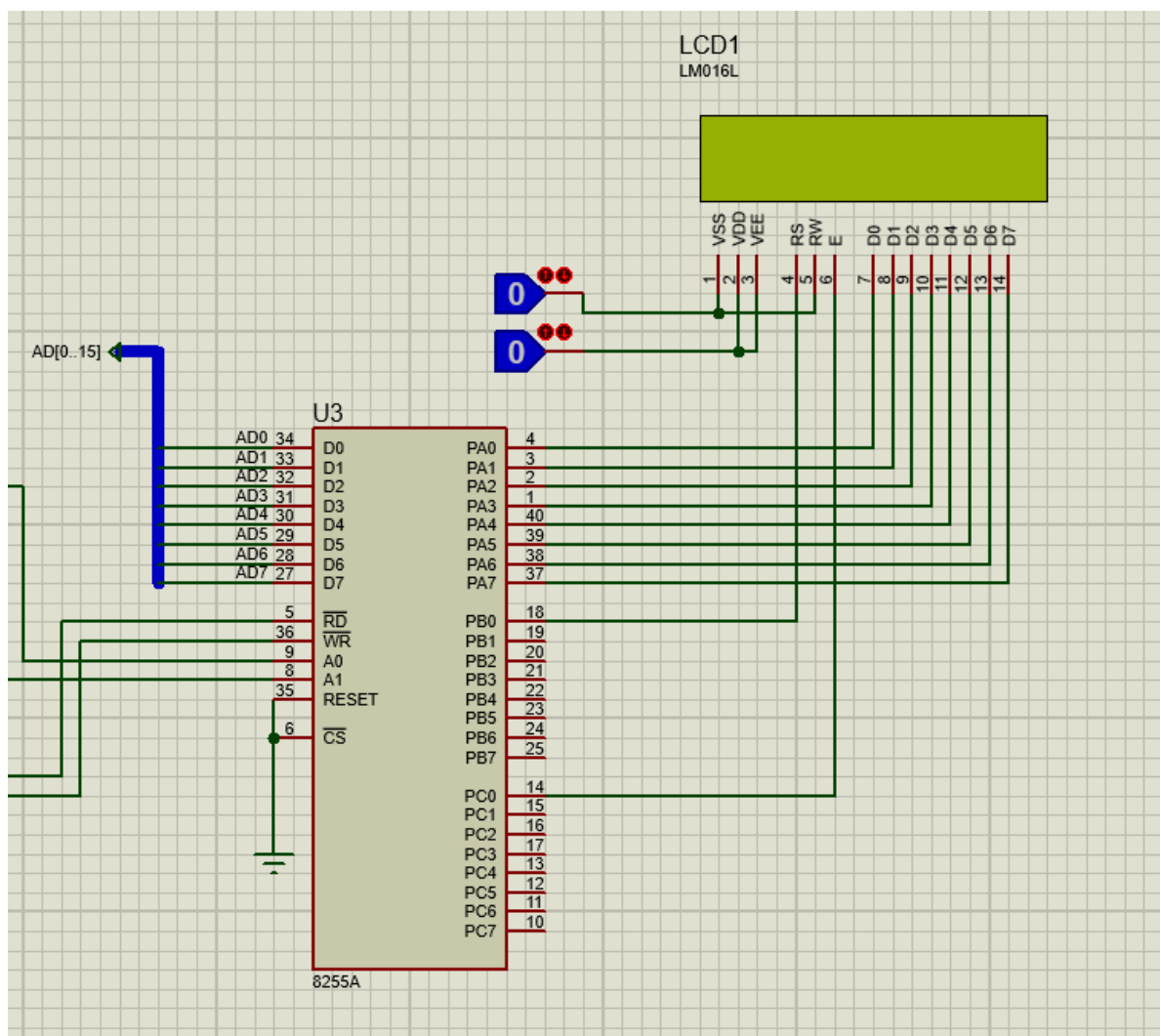
Dùng cổng A (Port A) để xuất ra dữ liệu và LCD hiển thị => MODE 0.

PA0 – PA7: Port A. Kết nối các tín hiệu ra với LCD 16x2 tương ứng. Xuất ra tín hiệu dựa trên tín hiệu vào hiển thị LCD 16x2.

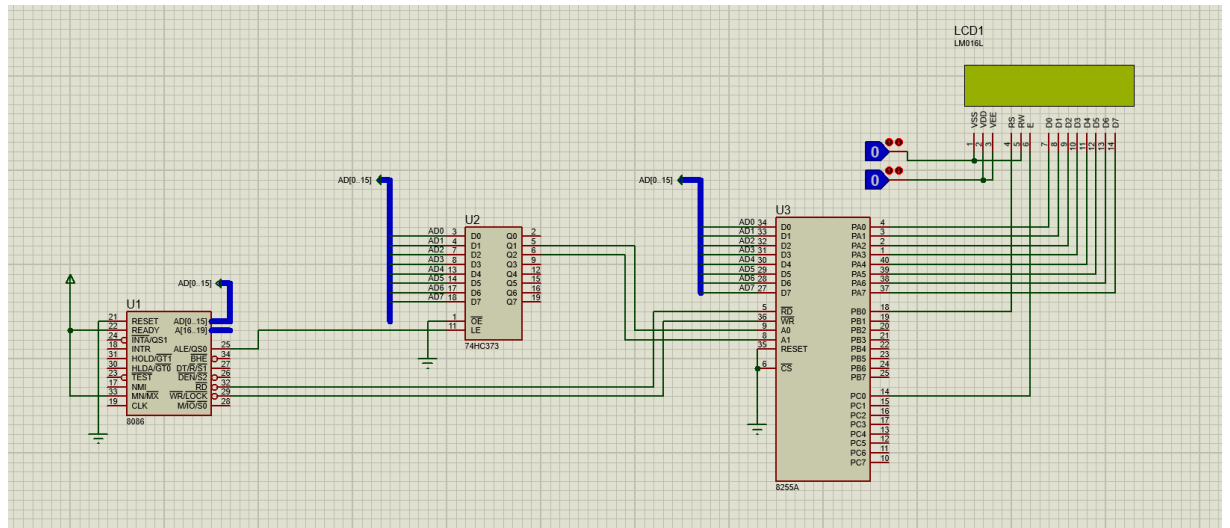
PB0: Được kết nối với chân RS.

PC0: Được kết nối với chân E.

### Sơ đồ kết nối LCD 16x2 và 8255A.



**Sơ đồ kết nối mạch.**



## Chương 2. NỘI DUNG CHÍNH

### 2.1. Giải thích code

Hàm khai báo cổng và vùng dữ liệu:

<pre>DATA SEGMENT PORTA EQU 00H PORTB EQU 02H PORTC EQU 04H PORT_CON EQU 06H DATA ENDS</pre>	<p>Khai báo vùng dữ liệu</p> <p>Địa chỉ Port A: 00H – địa chỉ giao tiếp, truyền dữ liệu với LCD.</p> <p>Địa chỉ Port B: 02H – địa chỉ chân RS, chọn thanh ghi lệnh hoặc thanh ghi dữ liệu.</p> <p>Địa chỉ Port C: 04H – địa chỉ chân E, chân cho phép sẽ kích hoạt lệnh khi được kéo lên mức cao rồi đưa về mức thấp.</p> <p>Địa chỉ Port_CON: 06H – địa chỉ dùng để điều khiển các chức năng mạch.</p>
--	---

Khởi tạo dữ liệu

<pre>CODE SEGMENT MOV AX, DATA MOV DS, AX  ORG 0000H START:  MOV DX, PORT_CON MOV AL, 10000000B OUT DX, AL</pre>	<p>Chuyển dữ liệu khai báo vào thanh ghi AX.</p> <p>Chuyển dữ liệu vào thanh ghi đoạn dữ liệu DS.</p> <p>Khởi tạo vùng địa chỉ ban đầu là 0000H.</p> <p>Chuyển địa chỉ PORT_CON vào DX để điều khiển chức năng.</p> <p>Chuyển 10000000B vào thanh ghi AL, tương đương 80H để cấu hình cho vị trí đầu tiên.</p> <p>Xuất dữ liệu tới cổng I/O bằng lệnh OUT.</p>
--	--

## Hàm cấu hình con trỏ

<pre>MOV AL,00H ;command setting cursor MOV DX,PORTC OUT DX,AL MOV AL,0EH MOV DX,PORTA OUT DX,AL MOV CX,00FFH ; Delay Delay1: LOOP Delay1</pre>	<p>Chuyển 00H vào AL.</p> <p>Chuyển địa chỉ PORTC vào DX.</p> <p>Lưu giá trị và xuất ra chân I/O nhằm mục đích tắt chân E rồi kéo lên mức cao sau khi truyền lệnh để thực hiện các lệnh cho LCD.</p> <p>Chuyển 0EH đến PORTA để thực hiện lệnh 0EH(bật hiển thị, con trỏ nhấp nháy).</p> <p>Thanh ghi CX là thanh ghi đếm, truyền 00FFH(255) vào thanh ghi này để thực hiện giảm CX từ giá trị 255 về 0 để tạo thành delay đảm bảo các câu lệnh được thực hiện hoàn toàn.</p>
<pre>MOV AL,0FFH MOV DX,PORTC OUT DX,AL MOV CX,00FFH ; Delay Delay2: LOOP Delay2</pre>	<p>Chuyển 0FFH tới địa chỉ PORT C để kéo chân E lên mức cao, kích hoạt lệnh 0EH sau đó delay để đảm bảo lệnh thực hiện hoàn toàn. Với các lệnh tiếp theo ta thực hiện bật tắt chân E tương tự.</p>

## Cấu hình 8bit cho LCD

<pre> MOV AL,00H ;command setting 8 bit mode MOV DX,PORTC OUT DX,AL MOV AL,038H MOV DX,PORTA OUT DX,AL MOV CX,00FFH ; Delay Delay3: LOOP Delay3 MOV AL,0FFH MOV DX,PORTC OUT DX,AL MOV CX,00FFH ; Delay Delay4: LOOP Delay4  MOV AL,00H ;command setting 8 bit mode MOV DX,PORTC OUT DX,AL MOV AL,0FH ;command setting 8 bit mode MOV DX,PORTB OUT DX,AL </pre>	<p>Tắt chân E để thực hiện câu lệnh tiếp.</p> <p>Chuyển 038H( lệnh cấu hình hiển thị 2 hàng và các ma trận 5x8).</p> <p>Delay để đảm bảo câu lệnh thực hiện hoàn toàn.</p> <p>Chuyển 0FFH tới PORTC để bật chân E.</p> <p>Tắt chân E để thực hiện câu lệnh tiếp theo.</p> <p>Chuyển 0FH vào PORTB (địa chỉ chân RS) để chọn chế độ thanh ghi dữ liệu để bắt đầu cấu hình 8bit hiển thị cho LCD.</p>
---	---

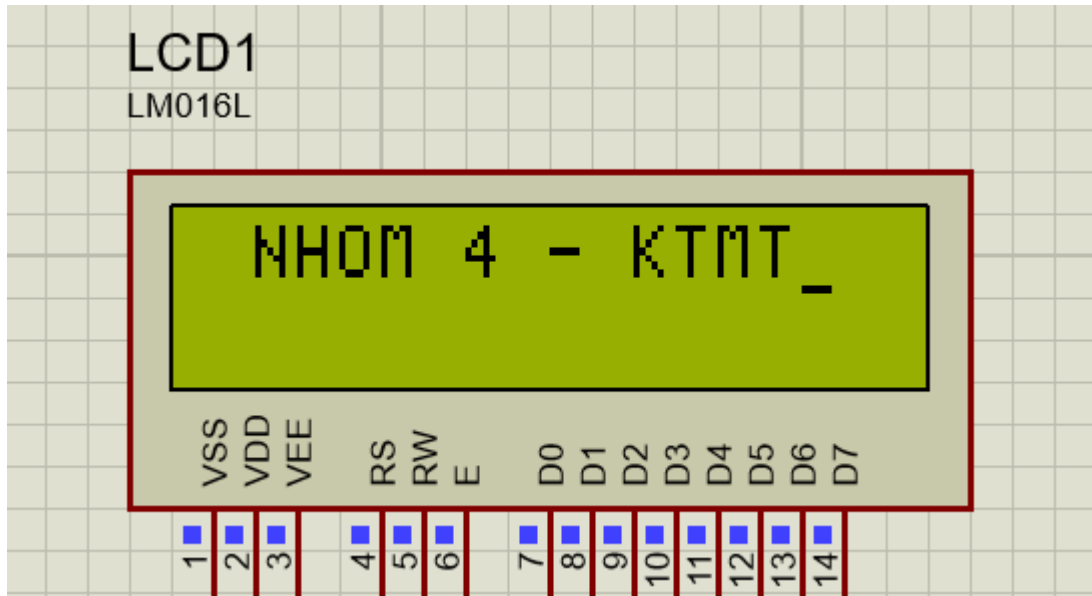
## In nội dung ra màn hình

<pre> MOV AL,'N' MOV DX,PORTA OUT DX,AL MOV CX,00FFH ; Delay Delay7: LOOP Delay7 MOV AL,0FFH MOV DX,PORTC OUT DX,AL MOV CX,00FFH ; Delay Delay8: LOOP Delay8 MOV AL,00H MOV DX,PORTC OUT DX,AL </pre>	<p>Để hiển thị kí tự, ta truyền vào địa chỉ PORT A rồi bật tắt chân E để thực hiện lệnh, thêm delay để đảm bảo lệnh thực hiện hoàn toàn.</p> <p>Kí tự 'N' sẽ được chuyển thành mã hex rồi hiển thị lên LCD, các kí tự khác cũng sẽ được chuyển đổi.</p>
---	---

## Chương 3. KẾT QUẢ

### 3.1. Kết quả mô phỏng

Sau khi nạp code ta chạy mô phỏng trên phần mềm “Proteus” ta được kết quả:



### 3.2. Trả lời câu hỏi bổ sung

**Phân tích vai trò:**

#### 1. AD0-AD15 của 8086:

Chân AD0-AD15 của 8086 là bus đa năng:

- Trong chu kỳ địa chỉ: Chúng mang địa chỉ (A0-A15).
- Trong chu kỳ dữ liệu: Chúng mang dữ liệu (D0-D15).

Do đó, cần dùng chốt (74HC373) để tách địa chỉ thấp (A0-A7) khi 8086 chuyển sang truyền dữ liệu.

#### 2. 74HC373:

D0-D7: Nhận tín hiệu địa chỉ thấp (AD0-AD7) từ 8086 trong chu kỳ địa chỉ và chốt giá trị này (khi ALE được kích hoạt).

Ngõ ra Q0-Q7: Xuất địa chỉ thấp (A0-A7) cho các thiết bị ngoại vi (bao gồm 8255A).

### **3. 8255A:**

D0-D7: Là bus dữ liệu 8-bit, kết nối trực tiếp với AD0-AD7 của 8086 trong chu kỳ dữ liệu.

A0, A1: Sử dụng 2 bit thấp nhất của địa chỉ (A0, A1) để chọn cổng giao tiếp hoặc thanh ghi điều khiển.

#### **Đầu nối chi tiết:**

##### **1. AD0-AD7 của 8086:**

###### **Đến 74HC373 (D0-D7):**

- Chân AD0-AD7 được nối trực tiếp với D0-D7 của 74HC373.
- Mục đích: Trong chu kỳ địa chỉ, AD0-AD7 mang địa chỉ thấp (A0-A7) và được chốt lại bởi 74HC373 khi ALE kích hoạt.
- Ngõ ra Q0-Q7 của 74HC373 sẽ nối với A0-A7 của 8255A.

###### **Đến 8255A (D0-D7):**

- Chân AD0-AD7 cũng được nối trực tiếp với D0-D7 của 8255A.
- Mục đích: Trong chu kỳ dữ liệu, AD0-AD7 mang dữ liệu (D0-D7) để giao tiếp với 8255A.

##### **2. AD8-AD15 của 8086:**

Không kết nối với 74HC373.

- Vì 74HC373 chỉ sử dụng để chốt địa chỉ thấp (AD0-AD7).

Đến 8255A:

- Chân AD8-AD15 không nối trực tiếp với D0-D7 của 8255A.

- Tuy nhiên, chúng có thể sử dụng để truyền địa chỉ cao hơn nếu cần, nhưng trong trường hợp này, chúng không được dùng làm dữ liệu.

### **Tín hiệu ALE và điều khiển:**

ALE của 8086: Nối với chân LE (Latch Enable) của 74HC373 để kích hoạt chốt địa chỉ thấp.

### **Lý do kết nối:**

#### **1. Chốt địa chỉ thấp:**

Do AD0-AD7 vừa mang địa chỉ, vừa mang dữ liệu, cần dùng 74HC373 để tách địa chỉ thấp ra trước khi bus chuyển sang truyền dữ liệu.

#### **2. Dữ liệu trực tiếp:**

D0-D7 của 8255A được nối trực tiếp với AD0-AD7 của 8086 để trao đổi dữ liệu 8-bit, phù hợp với thiết kế của 8255A.

#### **3. Đơn giản hóa địa chỉ hóa:**

Địa chỉ thấp (A0-A7) được xử lý bởi 74HC373, giúp giữ ổn định địa chỉ khi 8086 chuyển sang truyền dữ liệu.