

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

---



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

**Nhóm: LỚP N02 – NHÓM 06**

**Tên các thành viên**

<b>Tên thành viên</b>	<b>Mã số sinh viên</b>
Phạm Thị Thanh Hằng	B21DCCN328
Nguyễn Công Huân	B21DCCN403
Nguyễn Quang Thắng	B21DCCN669
Nguyễn Đình Văn	B21DCCN784
Hoàng Minh Tú	B21DCCN122

**Giảng viên hướng dẫn:** ThS. Đinh Xuân Trường

**Hà Nội 2023**

## **MỤC LỤC**

0. Thành viên và đóng góp	3
1. Giới thiệu đề tài	4
2. Nội dung chính của đề tài	4
3. Miêu tả chương trình	12
4. Miêu tả giao diện chương trình	12
5. Tài liệu tham khảo	14

## 0. Thành viên và đóng góp

STT	Mã sinh viên	Tên thành viên	% Đóng góp
1	B21DCCN328	Phạm Thị Thanh Hằng	24%
2	B21DCCN403	Nguyễn Công Huân	19%
3	B21DCCN669	Nguyễn Quang Thắng	19%
4	B21DCCN784	Nguyễn Đình Văn	19%
5	B21DCCN122	Hoàng Minh Tú	19%

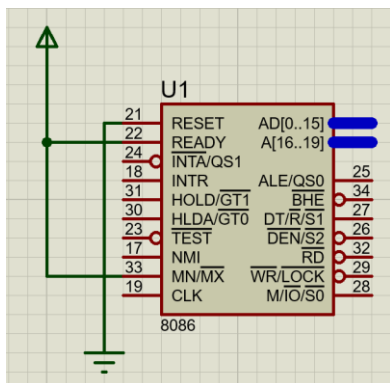
# Đề tài: Mô phỏng hiển thị đèn led 7 thanh để đếm số (từ 0 - 9)

## 1. Đề tài:

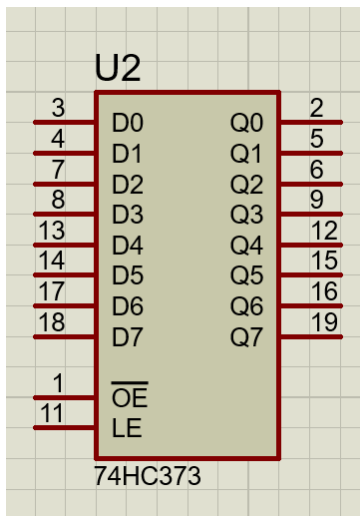
- Mô phỏng hiển thị đèn led 7 thanh để đếm số từ 0 đến 9

## 2. Nội dung tóm tắt chính:

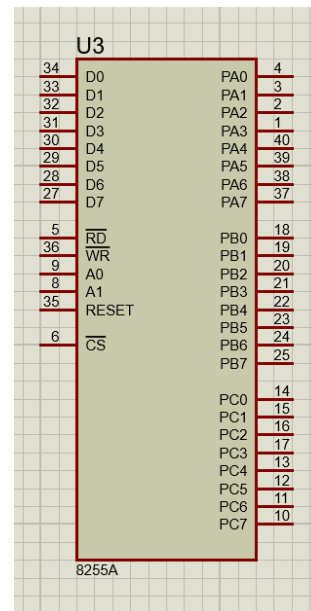
- Nối các cực dương (Common Anode) để dựa trên cấu trúc của chip thị LED 7 đoạn để LED biểu diễn các số 0 – 9 qua hệ nhị phân
- Sử dụng CPU 8086, 74HC373, 8255A
- Trình bày nội dung thanh ghi điều khiển chế độ của 8255A
- Trình bày cách hoạt động của mạch
- Trình bày, giải thích mã nguồn chương trình



Hình 1

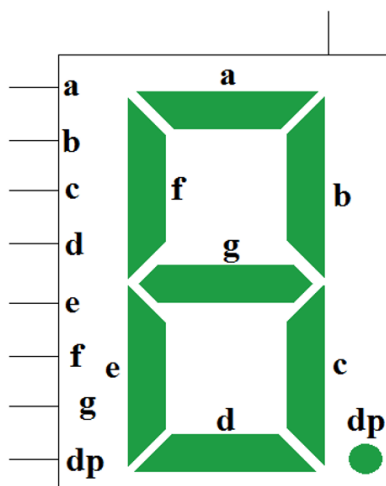


Hình 2



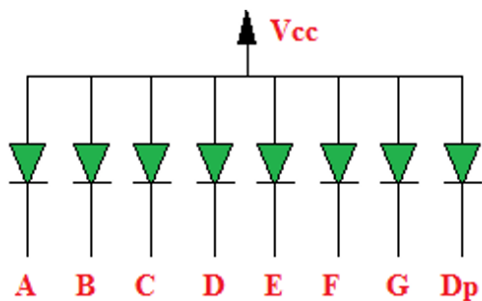
Hình 3

Hình 1: CPU 8086, Hình 2: 74HC373, Hình 3: 8255A



Hình 4

Hình 4. Cấu trúc của  
chỉ thị số 7 đoạn



Hình 5

Hình 5. LED Anode Chung

- Dp (Decimal Point): được sử dụng khi hiển thị số không phải là số nguyên

=> Đếm từ 0 – 9 là số nguyên => DP không hiển thị => Bit thứ 7 = 1

- Dựa vào các cực được nối, có thể phân loại LED 7 đoạn như sau:

+ Loại dương chung (Common Anode): nếu cực dương (anode) của tất cả 8 LED được nối với nhau và các cực âm (cathode) đứng riêng lẻ.

+ Loại âm chung (Common Cathode): nếu cực âm (cathode) của tất cả 8 LED được nối với nhau và các cực dương (anode) đứng riêng lẻ.

=> **Bài làm ví dụ cho loại dương chung (Common Anode)**

Xét mạch giải mã có các đầu ra tích cực ở mức thấp, dùng LED Anode chung

=> Thanh LED sáng: đầu ra tại bit đó = 0

tối: đầu ra tại bit đó = 1

Số	Số nhị phân								HEX
	7	6	5	4	3	2	1	0	
	dp	g	f	e	d	c	b	a	
0	1	1	0	0	0	0	0	0	C0
1	1	1	1	1	1	0	0	1	F9
2	1	0	1	0	0	1	0	0	A4
3	1	0	1	1	0	0	0	0	B0
4	1	0	0	1	1	0	0	1	99
5	1	0	0	1	0	0	1	0	92
6	1	0	0	0	0	0	1	0	82
7	1	1	1	1	1	0	0	0	8F
8	1	0	0	0	0	0	1	1	80
9	1	0	0	0	1	1	1	0	90

**Hình 6. Biểu diễn các số từ 0 – 9 theo hệ nhị phân và Hex dựa trên led 7 đoạn**

**- RESET của CPU 8086 được nối với chân đất**

+ RESET[I]: 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 và READY được nối với nguồn điện**

- 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

=> Ý tưởng theo chế độ MIN:

- WR[O] (WRITE): Xung cho phép ghi. Khi CPU đưa ra WR = 0 thì trên bus dữ liệu đã ổn định và chúng sẽ được ghi vào bộ nhớ hoặc ghi vào thiết bị ngoại vi khi WR = 1.
- RD[O] (READ): Xung cho phép đọc. Khi RD = 0 thì bus dữ liệu sẵn sàng nhận số liệu từ bộ nhớ hoặc thiết bị ngoại vi.

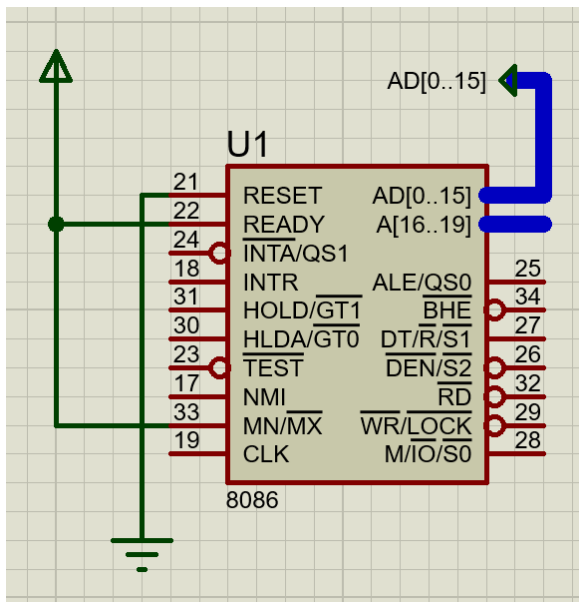
- ALE[O] (Address Latch Enable): Xung cho phép lưu trữ địa chỉ.

+ Khi 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).

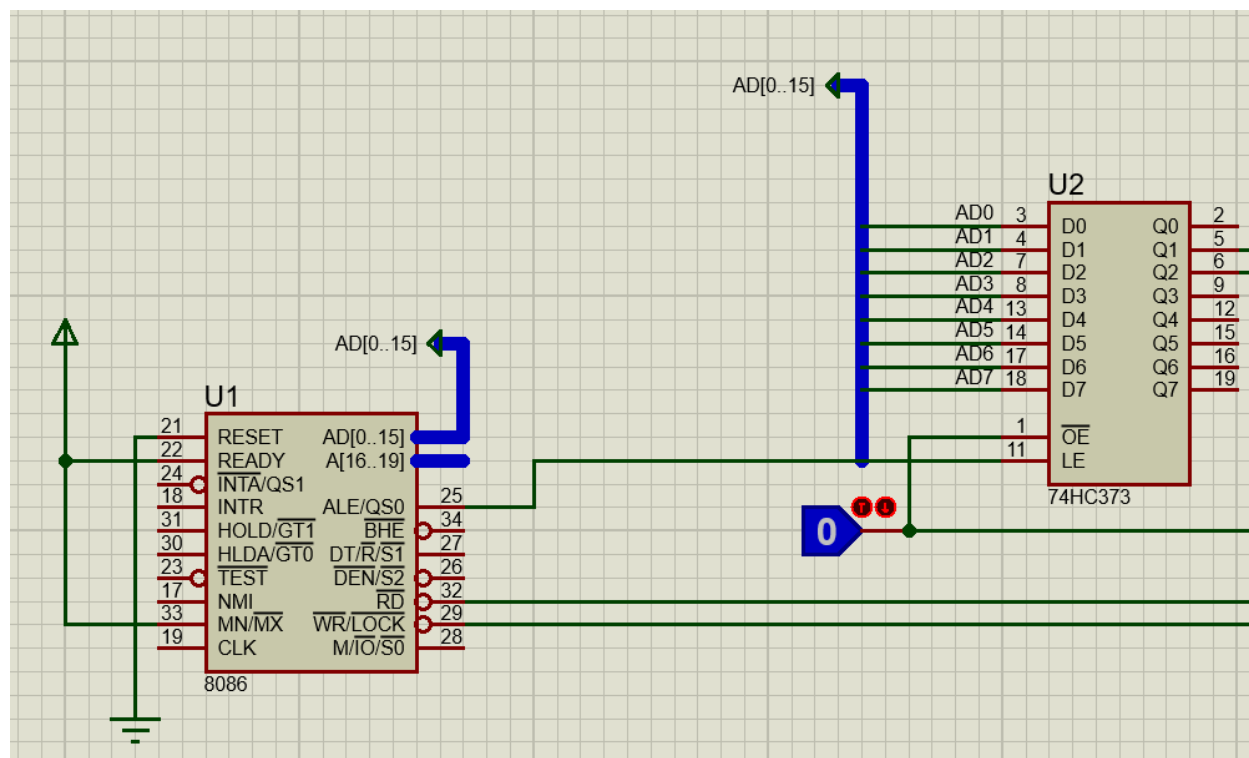
=> Các tín hiệu AD0 – AD15 được nối vào cùng 1 bus



**Hình 7. CPU 8086 khi đã  
nối RESET, READY, các bus**

- 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
- + Mức thấp => Mới hoạt động
- => Ý tưởng OE để ở mức thấp (OE = 0)

- 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



**Hình 8. Sau khi nối ALE của 8086 với LE của 74CH373,  
các tín hiệu đầu vào 74HC373 chung 1 bus**



- RESET: để khởi động 8255A khi cấp điện, khi bị Reset các thanh ghi, 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

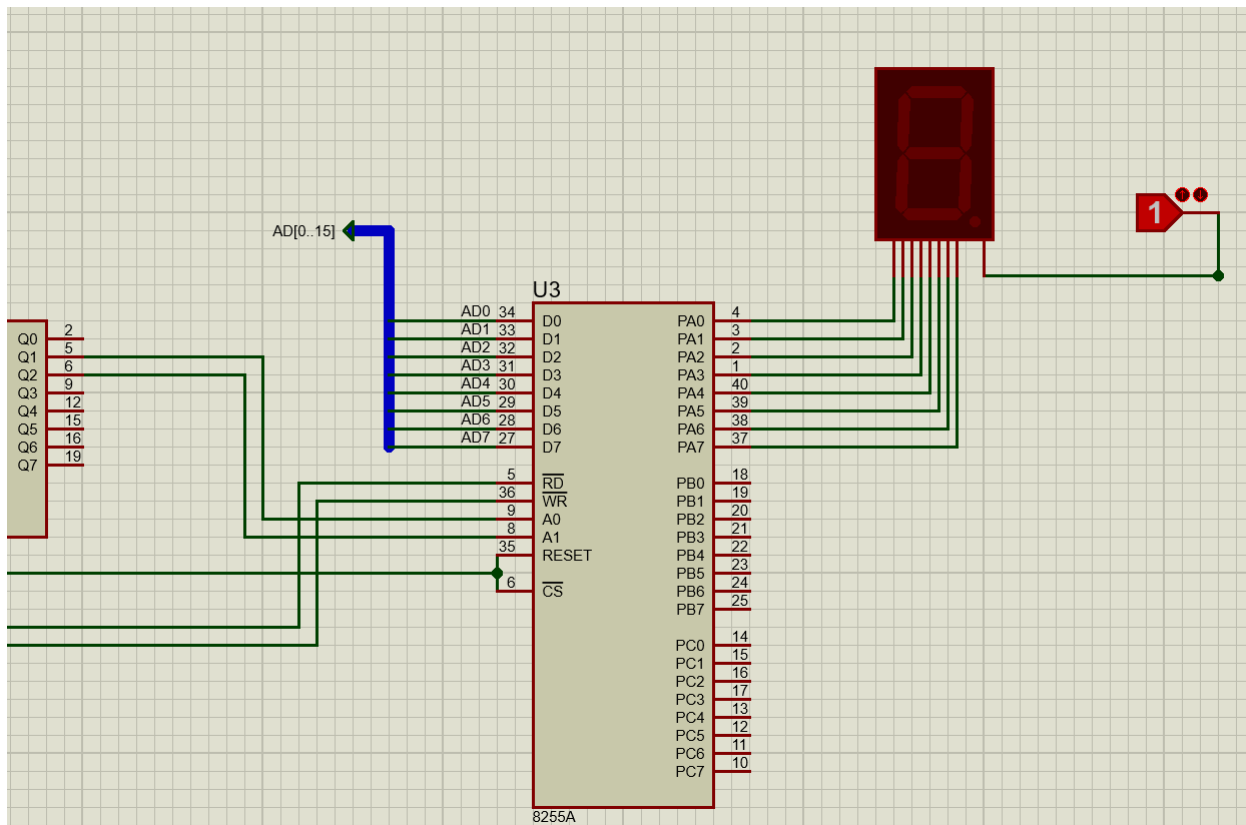
**RESET = 0**

- 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

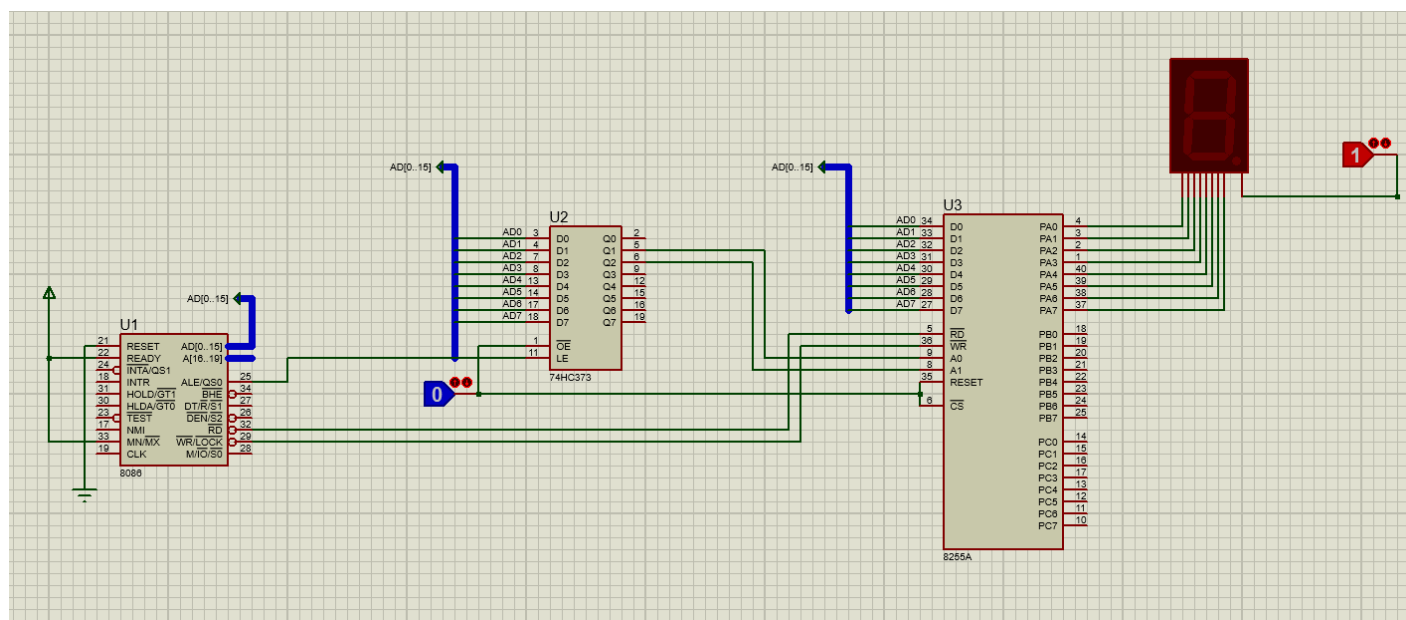
- RD và WR/LOCK của 8086 được nối với của 8255A

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



**Hình 9. Dữ liệu đầu ra cổng A kết nối với led 7 đoạn  
các tín hiệu đầu vào 8255A chung 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 đèn led hiển thị => MODE 0
- **PA0 – PA7: Port A. Kết nối các tín hiệu ra với các thanh led Cannode 7 đoạn tương ứng.** Xuất ra tín hiệu dựa trên tín hiệu vào hiển thị các thanh led sáng



Hình 10. Mạch đã hoàn chỉnh

\* Các thanh ghi của 8255A được các định qua tín hiệu địa chỉ  $A_0A_1$ :

A1	A0	Thanh ghi
X	X	Không sử dụng
0	0	Cổng A (PA)
0	1	Cổng B (PB)
1	0	Cổng C (PC)
1	1	Điều khiển

Thanh ghi điều khiển chế độ			
Từ điều khiển (Control word format)	D7	Xác lập chế độ	
	D6	Chọn chế độ: + 00: Chế độ 0 + 01: Chế độ 1	Nhóm cổng A
	D5	+ 02: Chế độ 2	
	D4	Cổng A: + 1 : Vào + 0 : Ra	
	D3	Cổng C (Cao): + 1 : Vào + 0 : Ra	
	D2	Chọn chế độ + 0: Chế độ 0 + 1: Chế độ 1	Nhóm cổng B
	D1	Cổng B: + 1 : Vào + 0 : Ra	
	D0	Cổng C (Thấp): + 1 : Vào + 0 : Ra	

Theo ý tưởng		
- Chế độ vào, ra	D7	1
- Chế độ 0	D6	00
	D5	
- Cổng A + Ra	D4	0
x	D3	0
x	D2	0
x	D1	0
x	D0	0
<b>Từ điều khiển (Control word format) = 1000000B = 80H</b>		

### 3. Miêu tả chương trình

**.MODEL SMALL**

**.STACK 100H**

**.DATA**

**PORTA EQU 00H** ; Địa chỉ cổng dữ liệu đầu ra A: 00H  
; Sử dụng để giao tiếp với các thiết bị ngoại vi đèn LED

**PORT\_CON EQU 06H** ; (Control Port): Địa chỉ cổng điều khiển - 06H  
; Điều khiển các chức năng của 8255A

**DELAY\_COUNT DW 1FFFFH** ; Điều chỉnh tốc độ hiển thị

**NUM DB 11000000B, 11111001B, 10100100B, 10110000B, 10011001B, 10010010B, 10000010B, 11011000B, 10000000B, 10010000B** ; Các số từ 0 - 9 hệ nhị phân  
; Bit thứ 0 đến 7 tương ứng với các đoạn LED a, b, c, d, e, f, g, dp  
; Bit = 1 : Tắt, Bit = 0 : Bật  
; Bit thứ 7 (dp): luôn tắt => Nhận giá trị 1

**.CODE**

**MAIN PROC**

**MOV AX, @DATA** ; Thiết lập DS để trỏ đến segment dữ liệu trong chương trình  
**MOV DS, AX**

**START:**

**MOV DX, PORT\_CON** ; Địa chỉ của cổng điều khiển (Port\_Control) được lưu vào DX  
; => DX = 0006H

**MOV AL, 80H** ; AL lưu từ điều khiển (Control word format)  
; như 1 thanh ghi điều khiển chế độ

**OUT DX, AL** ; Output từ AL xuất đến cổng

**MOV CX, 10** ; Số lần lặp tương ứng với số từ 0 đến 9

**MOV SI, OFFSET NUM** ; Lưu địa chỉ mảng NUM vào SI

### **LOOP\_DISPLAY:**

**MOV BX, DELAY\_COUNT** ; Đặt giá trị đếm cho vòng lặp delay vào BX

### **DELAY:**

**MOV AL, [SI]** ; Lấy bảng mã từng số

**MOV DX, PORTA** ; Địa chỉ cổng A được lưu vào DX

; để được điều khiển hiển thị các thanh LED

**OUT DX, AL** ; DX: Địa chỉ cổng A (PORT A):

; Từ AL hiển thị số qua đầu ra cổng A

; Khi này, giá trị tại AL là thể hiện các số từ 0- 9

**DEC BX**

**JNZ DELAY** ; Nhảy vào nhãn DELAY nếu khác 0 => Lặp 1FFFFH lần

**INC SI** ; Tăng SI để truy cập vào phần tử tiếp theo trong mảng

**LOOP LOOP\_DISPLAY** ; CX giảm dần, nếu CX != 0 thì chưa hiển thị đủ 10 số từ 0 – 9

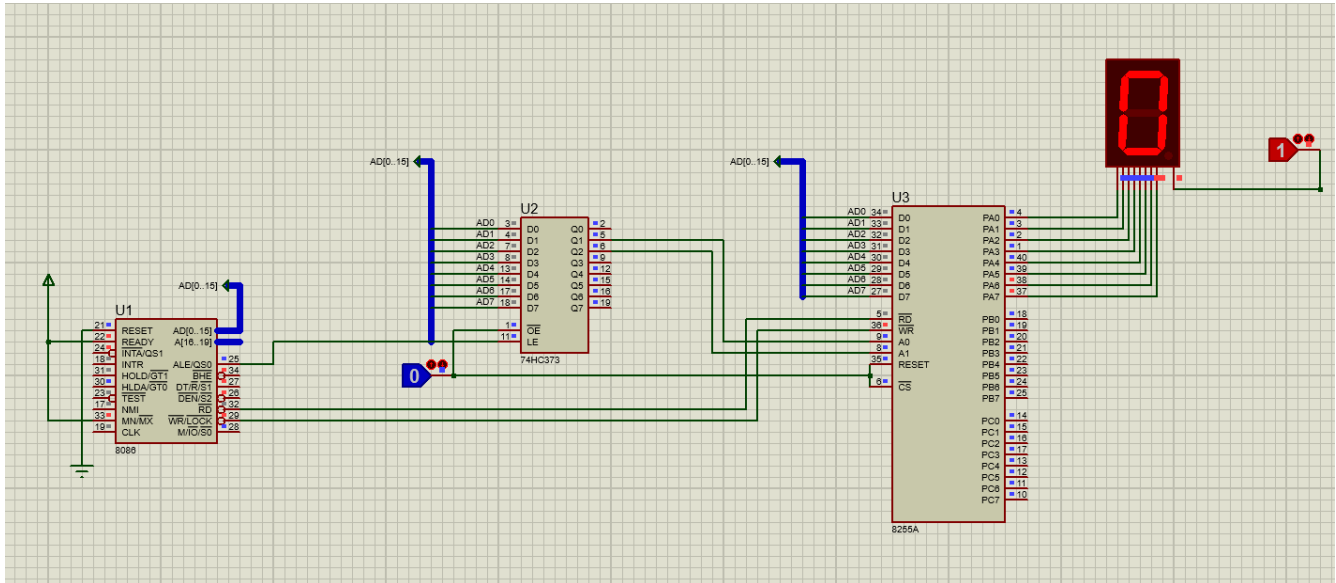
**JMP START** ; Lặp lại chương trình sau khi hiển thị đủ 0 - 9

**MAIN ENDP**

**END MAIN**

#### 4. Miêu tả giao diện chương trình

- Đèn LED 7 đoạn dựa vào đầu ra cổng A để tắt/ bật từ đó hiện ra số.



### Hình 11. Chương trình khi đếm số 0

- Sau 1 khoảng trễ nhất định, số 1 được chuyển thành số 1 và cứ thế khi đếm đến 9 thì lặp lại đếm từ 0

## 5. Các tài liệu tham khảo

- Tài liệu về 8086:

+ Datasheet:

[https://www.inf.pucrs.br/~calazans/undergrad/orgcomp\\_EC/mat\\_microproc/intel-8086\\_datasheet.pdf](https://www.inf.pucrs.br/~calazans/undergrad/orgcomp_EC/mat_microproc/intel-8086_datasheet.pdf)

- Tài liệu về 8255A:

+ Datasheet: <http://aturing.umcs.maine.edu/~meadow/courses/cos335/Intel8255A.pdf>

- Tài liệu về 74HC373:

+ Datasheet: [https://assets.nexperia.com/documents/data-sheet/74HC\\_HCT373.pdf](https://assets.nexperia.com/documents/data-sheet/74HC_HCT373.pdf)