

SỬ DỤNG PHẦN MỀM EMU 8086 VIẾT CHƯƠNG TRÌNH CHO VI XỬ LÝ 80X86

I. Mục tiêu bài học

Biết được một số phần mềm lập trình hợp ngữ, sử dụng thành thạo phần mềm EMU 8086 để soạn thảo, hợp dịch, sửa lỗi và mô phỏng cho VXL 80x86.

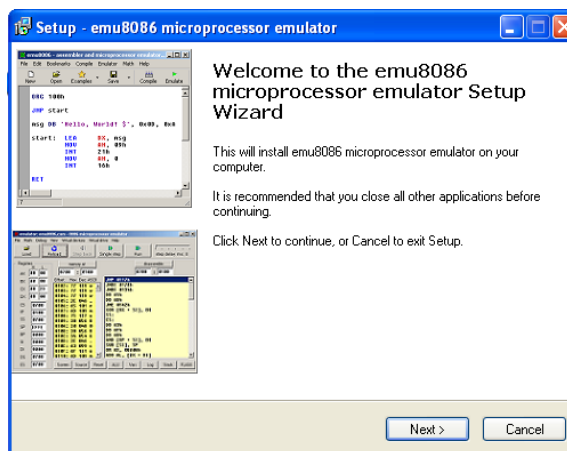
- Vận dụng làm các ví dụ, bài tập cho vi xử lý trên EMU 8086

II. Nội dung bài học

1. Hướng dẫn sử dụng phần mềm EMU 8086.

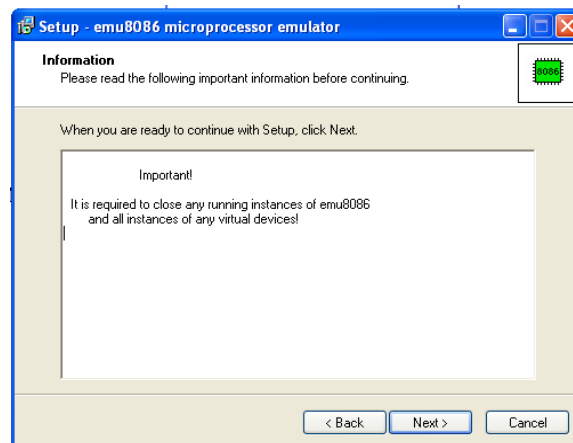
1.1 Hướng dẫn cài đặt phần mềm EMU 8086

- Nháy đúp chuột vào File có tên là Setup.exe có trong bộ cài EMU 8086. Một cửa sổ hiện lên thông báo chào mừng bạn đến với chương trình mô phỏng cho vi xử lý và báo cho bạn là chương trình đang chuẩn bị cài đặt, để tiếp tục bạn chọn *Next*



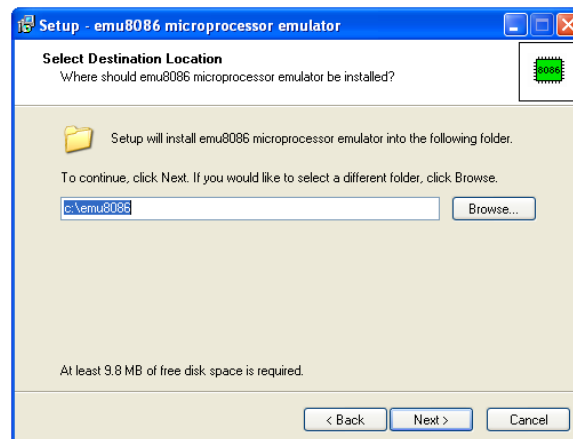
Hình 1 Cửa sổ chào mừng của phần mềm cài đặt EMU 8086

- Tiếp theo phần mềm đưa ra thông báo yêu cầu người sử dụng đọc kỹ trước khi tiến hành cài đặt tiếp, để tiếp tục bạn chọn Next



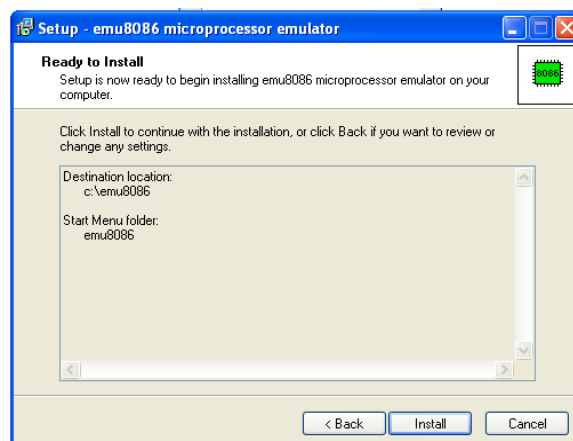
Hình 2 Các thông tin sử dụng phần mềm

- Hộp thoại tiếp theo thông báo cho người cài đặt thiết lập đường dẫn cho thư mục mà chương trình sẽ cài đặt ra, bạn nên để mặc định và chọn Next.



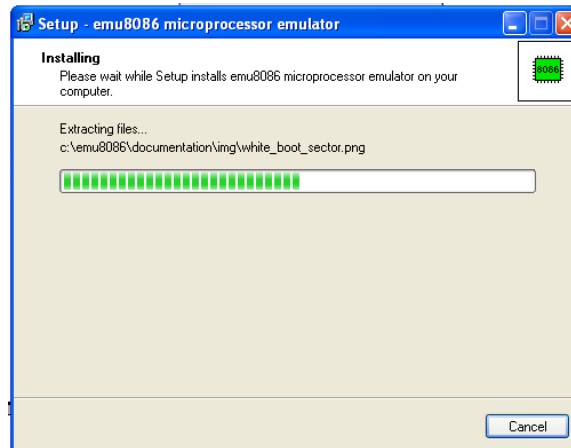
Hình 3 Thông báo lựa chọn đường dẫn cài đặt

- Hộp thoại kế tiếp báo chương trình đã sẵn sàng cài đặt, bạn tiếp tục chọn Install



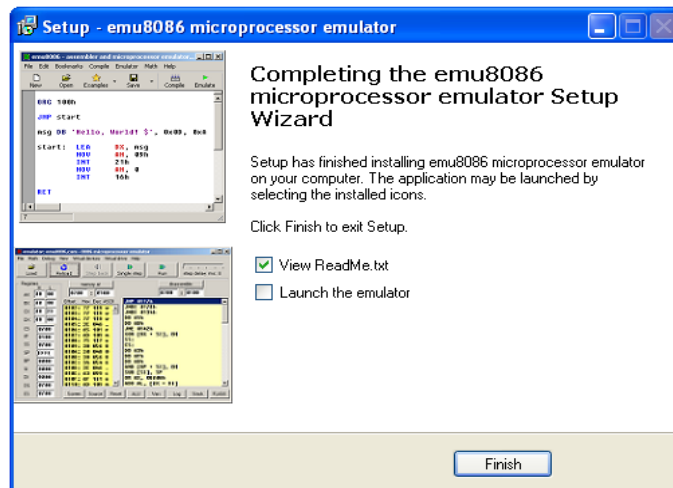
Hình 4 Các thông tin cài đặt

- Khi đó chương trình sẽ tiến hành cài đặt lên máy bạn.



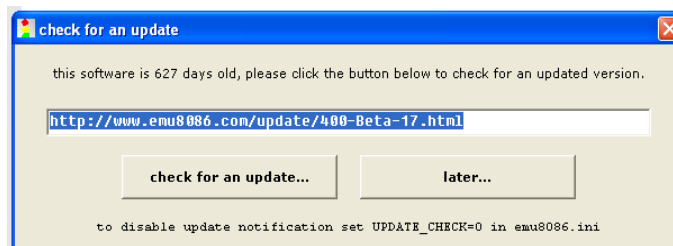
Hình 5 Tiến trình cài đặt

- Sau khi cài đặt xong ch- ơng trình hiện lên hộp thoại thông báo quá trình cài đặt đã hoàn tất, để kết thúc chọn Finish.



Hình 6 Thông báo kết thúc quá trình cài đặt

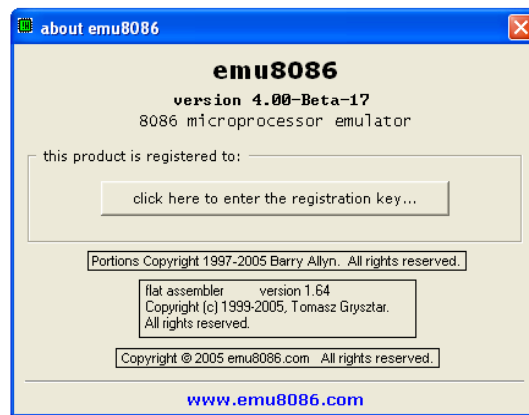
- Bắt đầu sử dụng ch- ơng trình chọn: Start/Program/Emu8086/emu8086 khi đó ch- ơng trình hiện lên hộp thoại



Hình 7 Yêu cầu cập nhật phần mềm

Để tiếp tục chọn Later

- Đăng ký bản quyền phần mềm chọn About trên thanh công cụ, khi đó ch- ơng trình hiện lên hộp thoại



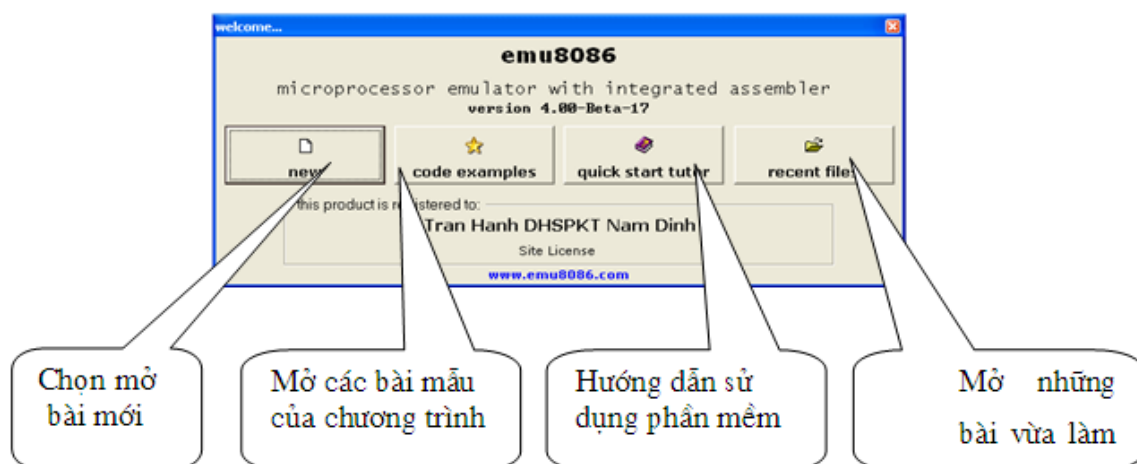
Hình 8 Thông tin bản quyền phần mềm

- Để nhập CD key bạn nháy chuột vào “Click here to enter the registration key...” lúc này chương trình đưa ra hộp thoại với hai dòng để nhập tên người sử dụng và mã bản quyền.
- Khi đăng ký hoàn tất chương trình đưa lên hộp thoại cho biết phần mềm đã được đăng ký cũng như phiên bản của phần mềm.

1.2. Sử dụng phần mềm EMU 8086

1.2.1. Khởi động chương trình và các menu của chương trình.

- Khởi động chương trình chọn từ biểu tượng gọi tắt trên màn hình hoặc từ Start/Programs/EMU8086/EMU8086, khi đó chương trình hiện lên hộp thoại cho phép lựa chọn như sau.

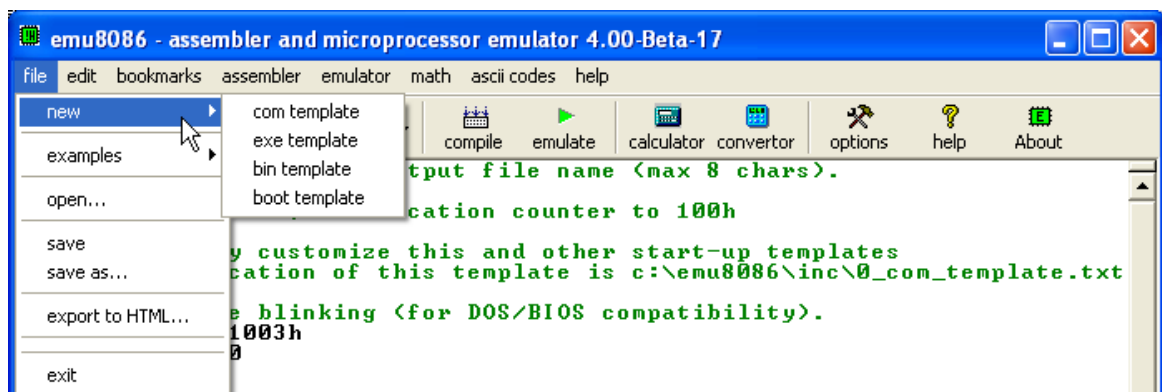


Hình 9 Các tùy chọn làm việc với EMU 8086

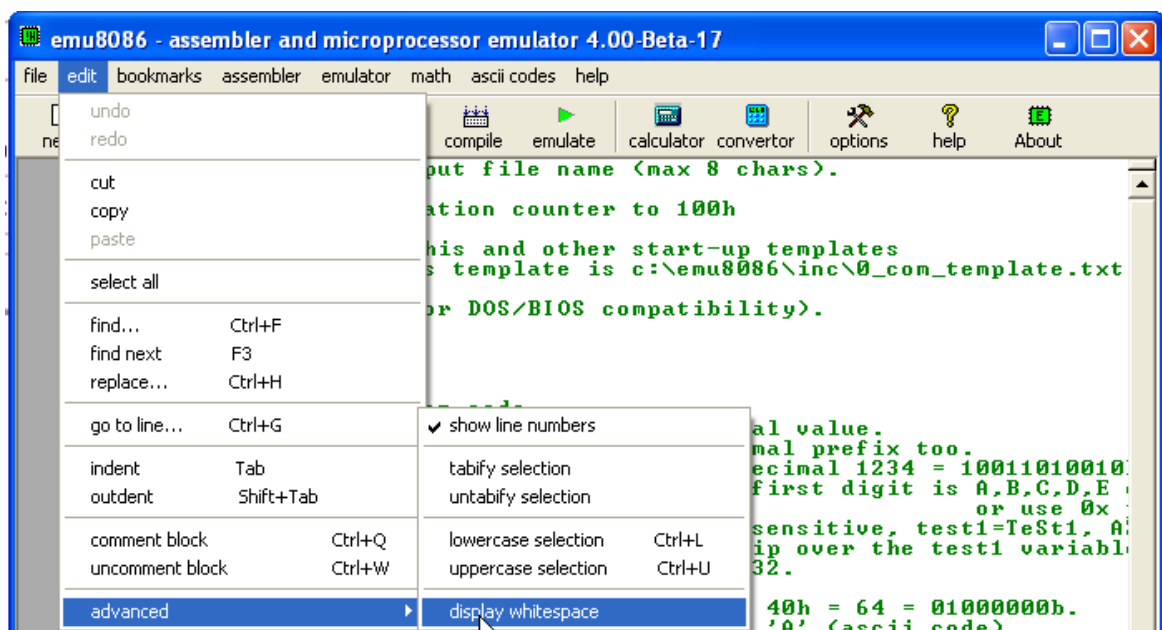
Các menu chính của chương trình.

- Menu File:

- + New : cho phép mở bài mới dạng *.COM, *.EXE, dạng Bin và dạng Boot.
- + Examples : cho phép mở các ví dụ có sẵn của chương trình
- + Open : cho phép mở các bài tập đã được ghi vào trong máy hay trên đĩa.
- + Save : cho phép ghi lại chương trình hiện tại đang làm
- + Save as : cho phép ghi lại chương trình đang làm với tên khác hoặc đường dẫn khác đi.
- + Export to HTML : cho phép bạn xuất file của chương trình đang làm thành file có đuôi HTML.



- Menu Edit:

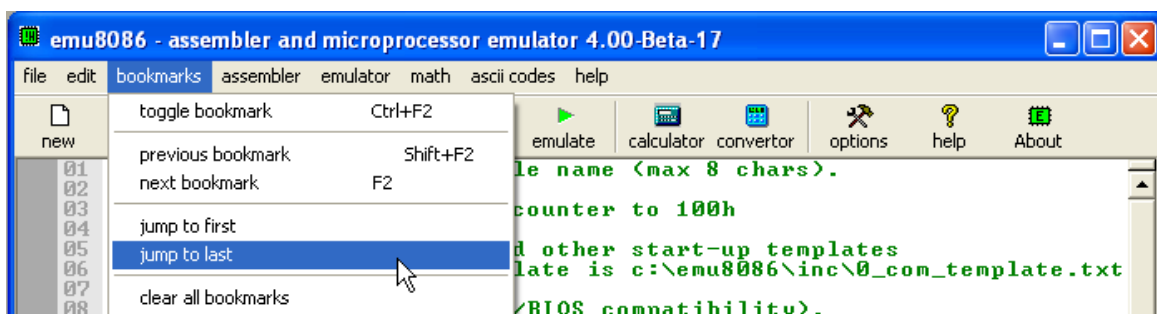


- + Undo: phục hồi lại một lệnh mà bạn vừa thao tác
- + Redo: phục hồi lại một lệnh vừa thao tác về phía trước

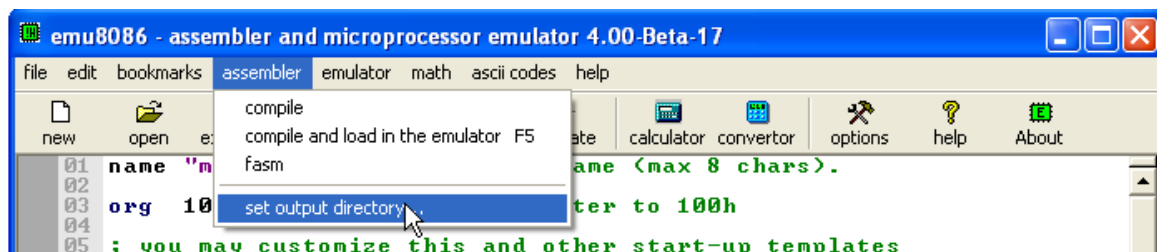
- + Cut: xoá một đoạn ch- ơng trình đ- ọc bôi đen
- + Copy: sao chép một đoạn ch- ơng trình đ- ọc bôi đen
- + Paste: dán lại phần vừa thực hiện lệnh copy hay lệnh cut
- + Select all: lựa chọn toàn bộ ch- ơng trình (đánh dấu)
- + Find: tìm kiếm ký tự hay đoạn văn bản trong ch- ơng trình
- + Find next: tìm kiếm các ký tự tiếp theo từ vị trí con trỏ hiện hành
- + Replace: tìm kiếm một hay nhiều ký tự và thay thế nó bởi ký tự mới
- + Go to line: đ- a con trỏ tới dòng đ- ọc nhập từ bàn phím

Ngoài ra trong tap menu này còn một số tính năng khác.

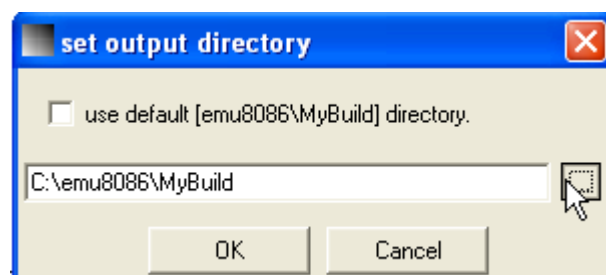
- Menu Bookmark: đánh dấu các đoạn ch- ơng trình đang viết.



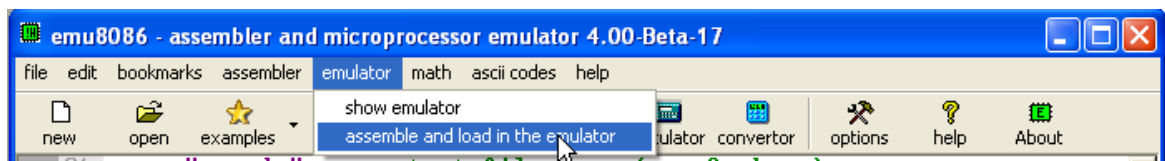
- Menu Assembler: chứa các tính năng sau



- + Compile: hợp dịch ch- ơng trình.
- + Compile and load in the emulator: hợp dịch ch- ơng trình và nạp file chạy của ch- ơng trình vào vùng mô phỏng.
- + Set output directory: đặt đ- ường dẫn cho đầu ra.



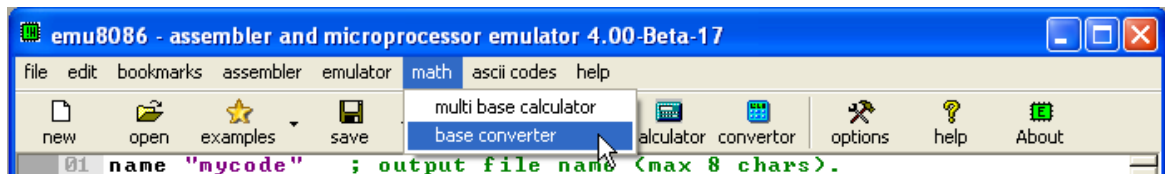
- Menu Emulator:



+ Show emulator : cho phép quan sát của sổ mô phỏng của ch-ơng trình

+ Assemble and load in the emulator: hợp dịch ch-ơng trình và tải ch-ơng trình vào vùng mô phỏng

- Menu math: cho phép thực hiện một số phép toán số học



+ Multi base calculator: cho phép quan sát kết quả của quá trình mô phỏng d-ới dạng các hệ đếm.

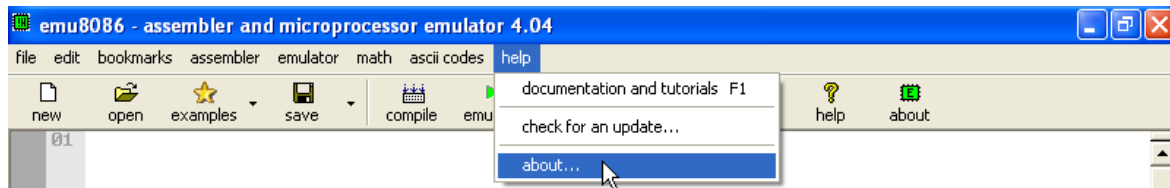
+ Base convertor : chuyển đổi các hệ đếm nh- hệ thập phân, nhị phân, hexa.

- Menu ASSCII codes: Cho phép quan sát mã ASSCII t-ơng ứng của các ký tự trên bàn phím máy tính và các ký tự đặc biệt khác. Bạn có thể quan sát mã của ký tự d-ới dạng số Hexa hay thập phân bằng cách nháy chuột ra màn hình để thay đổi.

Dạng Hexa:

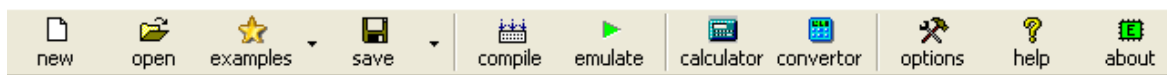
ascii codes															
00: null	20: spa	40: @	60: `	80: C	A0: á	C0: L	E0: α	01: �	21: !	41: A	61: a	81: �	A1: �	C1: �	E1: �
02: �	22: "	42: B	62: b	82: �	A2: �	C2: �	E2: �	03: �	23: #	43: C	63: c	83: �	A3: �	C3: �	E3: �
04: �	24: \$	44: D	64: d	84: �	A4: �	C4: �	E4: �	05: �	25: %	45: E	65: e	85: �	A5: �	C5: �	E5: �
06: �	26: &	46: F	66: f	86: �	A6: �	C6: �	E6: �	07: beep	27: '	47: G	67: g	87: �	A7: �	C7: �	E7: �
08: back	28: (48: H	68: h	88: �	A8: �	C8: �	E8: �	09: tab	29:)	49: I	69: i	89: �	A9: �	C9: �	E9: �
0A: newl	2A: *	4A: J	6A: j	8A: �	AA: �	CA: �	EA: �	0B: �	2B: +	4B: K	6B: k	8B: �	AB: �	CB: �	EB: �
0C: �	2C: ,	4C: L	6C: l	8C: �	AC: �	CC: �	EC: �	0D: �	2D: -	4D: M	6D: m	8D: �	AD: �	CD: �	ED: �
0E: �	2E: .	4E: N	6E: n	8E: �	AE: �	CE: �	EE: �	0F: �	2F: /	4F: O	6F: o	8F: �	AF: �	CF: �	EF: �
10: �	30: 0	50: P	70: p	90: �	B0: �	D0: �	F0: �	11: �	31: 1	51: Q	71: q	91: �	B1: �	D1: �	F1: �
12: �	32: 2	52: R	72: r	92: �	B2: �	D2: �	F2: �	13: �	33: 3	53: S	73: s	93: �	B3: �	D3: �	F3: �
14: �	34: 4	54: T	74: t	94: �	B4: �	D4: �	F4: �	15: �	35: 5	55: U	75: u	95: �	B5: �	D5: �	F5: �
16: �	36: 6	56: V	76: v	96: �	B6: �	D6: �	F6: �	17: �	37: 7	57: W	77: w	97: �	B7: �	D7: �	F7: �
18: �	38: 8	58: X	78: x	98: �	B8: �	D8: �	F8: �	19: �	39: 9	59: Y	79: y	99: �	B9: �	D9: �	F9: �
1A: �	3A: :	5A: Z	7A: z	9A: �	BA: �	DA: �	FA: �	1B: �	3B: ;	5B: [7B: {	9B: �	BB: �	DB: �	FB: �
1C: �	3C: <	5C: \	7C:	9C: �	BC: �	DC: �	FC: �	1D: �	3D: =	5D:]	7D: }	9D: �	BD: �	DD: �	FD: �
1E: �	3E: >	5E: ^	7E: ~	9E: �	BE: �	DE: �	FE: �	1F: �	3F: ?	5F: _	7F: �	9F: �	BF: �	DF: �	FF: res

- Menu Help:



Trợ giúp về cách sử dụng chương trình, tập lệnh 8086, các ngắt và một số bài tập mẫu.

b. Các thanh công cụ của chương trình.

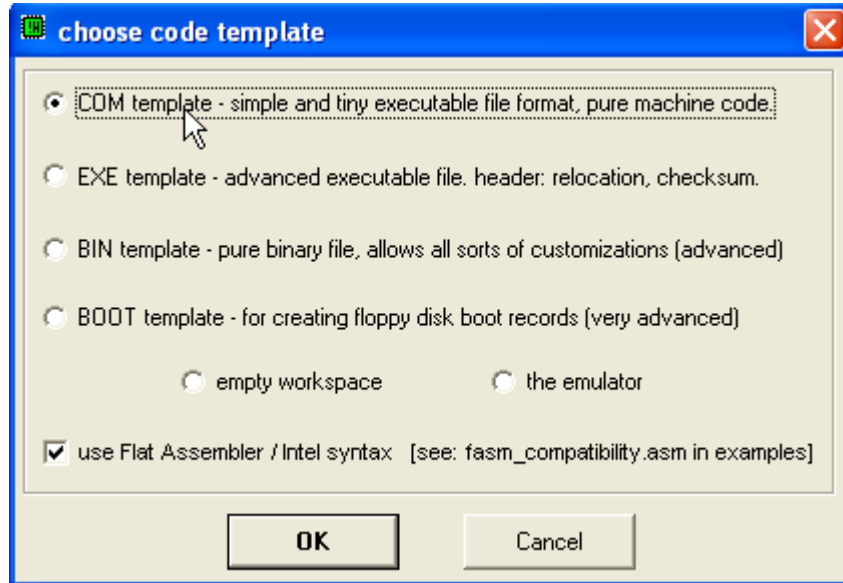


- + New: mở bài mới
- + Open: mở bài đã có
- + Examples: mở các bài ví dụ có sẵn của chương trình.
- + Save: ghi lại bài đang làm.
- + Compile: hợp dịch chương trình
- + Emulate: chuyển chương trình sang chế độ mô phỏng
- + Calculator: quan sát kết quả dưới dạng các hệ đếm khác nhau
- + Convertor: chuyển đổi các hệ đếm
- + Option: tùy chọn các tính năng ứng dụng của chương trình
- + Help: trợ giúp
- + About : cung cấp các thông tin cơ bản về phần mềm EMU 8086.

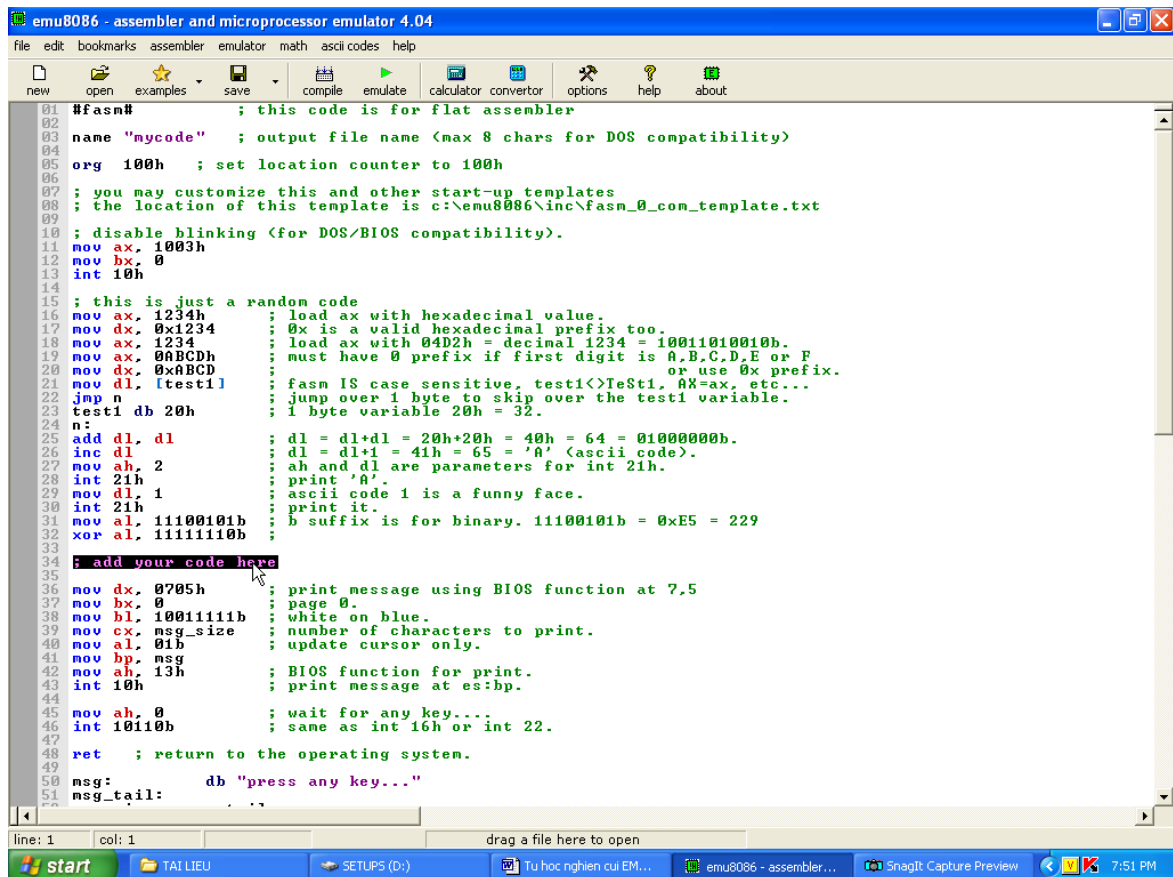
1.2.2. Soạn thảo chương trình

Chương trình cho vi xử lý 8086 ta có thể viết dưới hai dạng khung chương trình đó là khung chương trình dạng COM và khung chương trình dạng EXE. Vì vậy khi bắt đầu vào viết chương trình cho bất kỳ một ứng dụng nào phải lựa chọn khung chương trình. Nhắc lại là việc lựa chọn khung chương trình phụ thuộc vào dung lượng, quy mô (độ lớn) của chương trình. Nếu viết chương trình cho vi xử lý sử dụng một số phần mềm khác thì phải khai báo khung cho chương trình, phần mềm EMU 8086 đã tạo kết cấu có sẵn ứng với các khung chương trình này.

+ Chọn khung ch- ơng trình dạng COM (cho các ứng dụng nhỏ). Chọn mở bài mới và chọn vào tùy chọn COM.

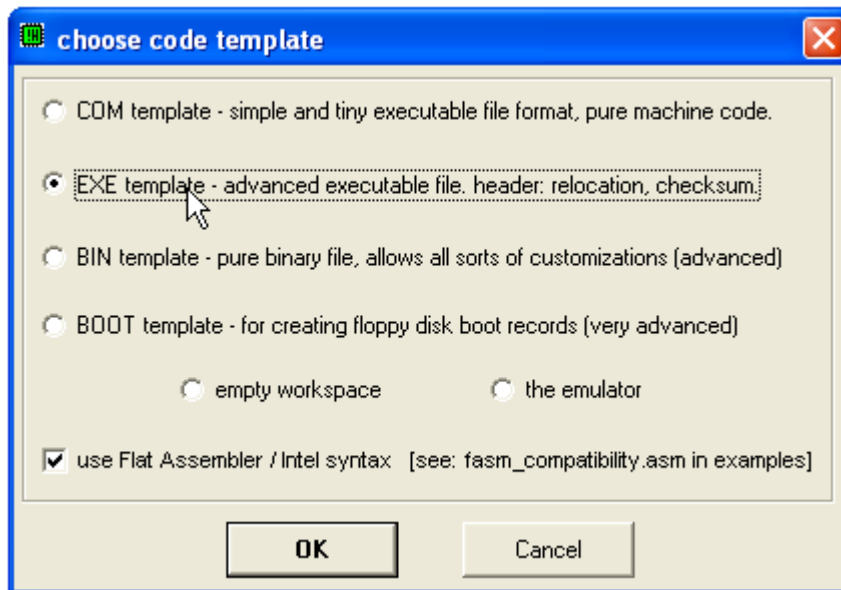


Khi đó ch- ơng trình sẽ cho ta khung của ch- ơng trình dạng COM nh- sau:



Viết nội dung ch- ơng trình chính vào vị trí đ- ợc bôi đen “Add your code here”

+ Chọn khung ch- ơng trình dạng EXE (cho các ứng dụng từ nhỏ tới lớn).
Chọn mở bài mới và chọn vào tùy chọn EXE.



Chương trình mở ra một file mới có giao diện như sau:

```
01 ; multi-segment executable file template.
02
03 ; flat assembler syntax
04
05 format MZ
06
07 entry code_seg:start ; set entry point
08
09 stack 256
10
11
12 segment data_seg
13 ; add your data here!
14 pkey db "press any key...$"
15
16
17
18 segment code_seg
19 start:
20 ; set segment registers:
21 mov ax, data_seg
22 mov ds, ax
23 mov es, ax
24
25 ; add your code here
26
27 mov dx, pkey
28 mov ah, 9
29 int 21h ; output string at ds:dx
30
31 ; wait for any key...
32 mov ah, 1
33 int 21h
34
35 mov ax, 4c00h ; exit to operating system.
36 int 21h
37
38
39
```

Viết nội dung chương trình chính vào vị trí được bôi đen “Add your code here”.

Sau khi đã có được lựa chọn về cấu trúc khung của chương trình hợp ngữ cần làm cho ứng dụng ta sẽ tiến hành soạn thảo chương trình cho VXL:

Khi tiến hành soạn thảo chương trình cần theo qui - ớc của lập trình hợp ngữ về cấu trúc của dòng lệnh, tên nhãn, lệnh, toán hạng, chú thích. Chương trình chính, chương trình con phải viết đúng cấu trúc.

L- u ý: Việc soạn thảo chương trình có thể viết hoa hoặc viết th - ờng nh - ng nên thống nhất một cách viết trên một bài, không nên sử dụng phong tiếng Việt trong lúc soạn thảo. Nên phân biệt giữa chương trình chính và chương trình con (bằng các chú thích, gạch nối, ...) để dễ cho việc quan sát và sửa lỗi.

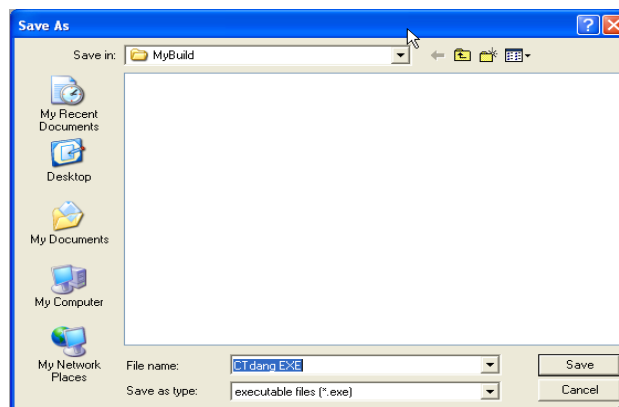
Tr- ớc khi tiến hành soạn thảo nên ghi lại chương trình để tránh các rủi ro do mất điện, treo máy. Việc ghi lại bằng cách chọn Save trên menu hay trên thanh công cụ. Mặc định đ - ờng dẫn của chương trình là C:\EMU8086\Mysource.

L- u ý: không nên đặt tên file dài quá 8 ký tự, không đ - ợc có dấu chấm, dấu gạch giữa các ký tự. Đây là qui - ớc của chương trình hợp dịch hợp ngữ.

1.2.3. Dịch và liên kết chương trình

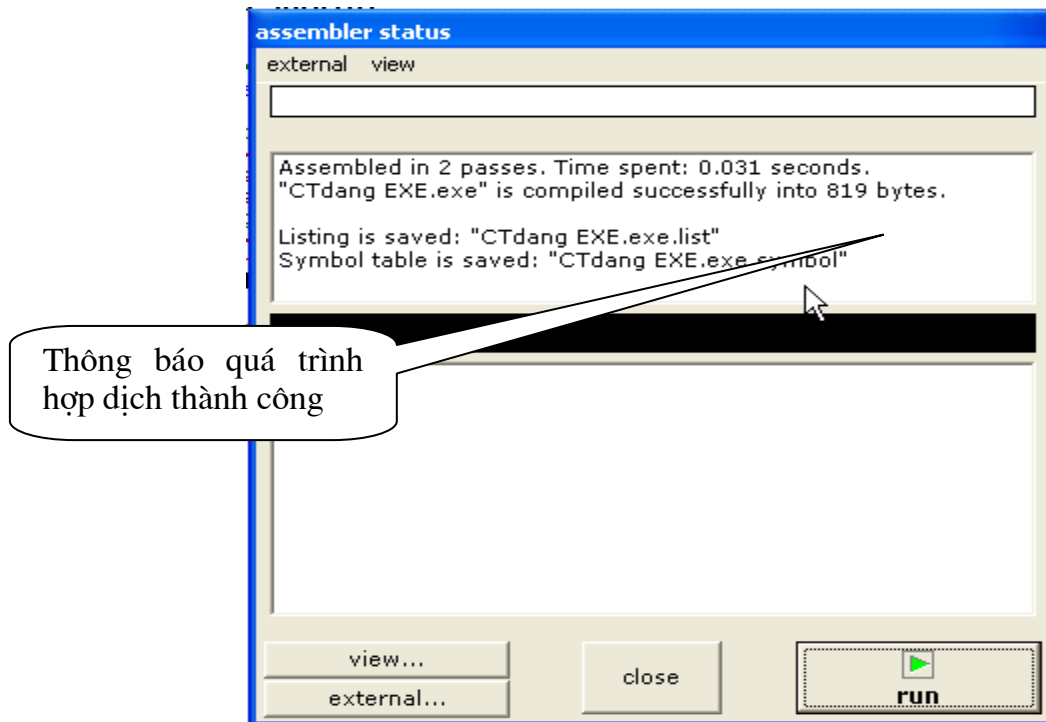
+ Hợp dịch chương trình:

Để tiến hành hợp dịch, trong Menu Assembler chọn compile trên thanh công cụ của chương trình. Khi đó chương trình đ - a ra hộp thoại yêu cầu ghi tên file chạy (.exe hoặc .com) mà trình hợp dịch sẽ tạo ra.



Chọn đ - ờng dẫn và đánh tên cho ứng dụng muốn tạo ra (có thể để đ - ờng dẫn mặc định của chương trình và tên trùng với tên mà tr- ớc đó đã đặt cho file nguồn).

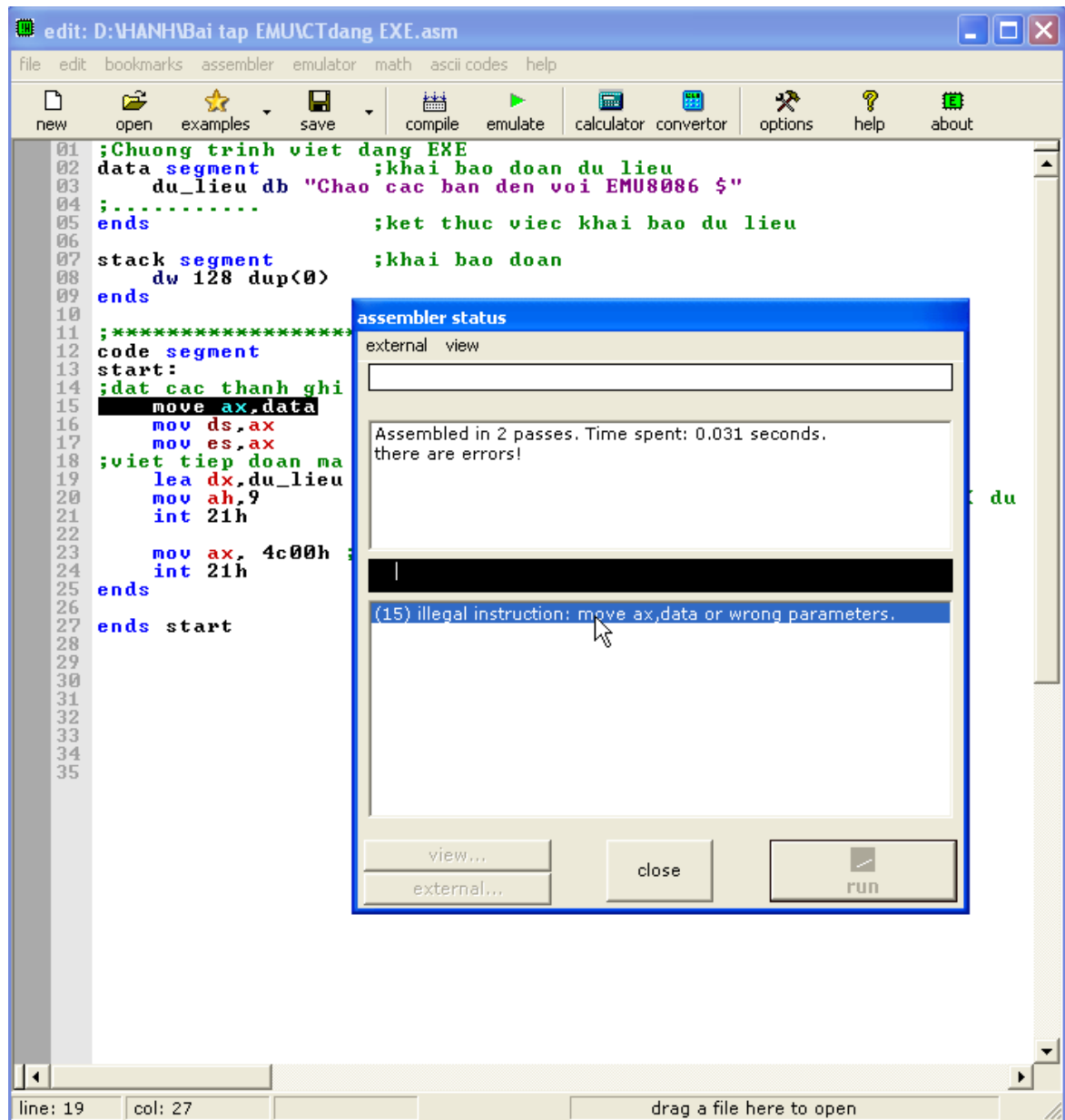
Nếu chương trình không có lỗi về cú pháp (lỗi đánh máy, sai cấu trúc dòng lệnh) thì máy tính sẽ đưa ra thông báo quá trình hợp dịch thành công và dung lượng chương trình được tạo ra.



+ Sửa lỗi : lập trình viên thường mắc phải hai dạng lỗi cơ bản đó là lỗi giải thuật và lỗi cú pháp.

Đối với lỗi cú pháp thì nguyên nhân của việc mắc các lỗi này là do đánh máy, sai cấu trúc câu lệnh, sai chú thích, sai cấu trúc chương trình, v.v. Mắc phải lỗi này, khi hợp dịch chương trình máy tính sẽ báo lỗi và dựa vào thông báo lỗi sẽ biết nguyên nhân và vị trí lỗi.

Ví dụ một chương trình viết sai cú pháp, khi hợp dịch chương trình máy tính đưa ra thông báo:



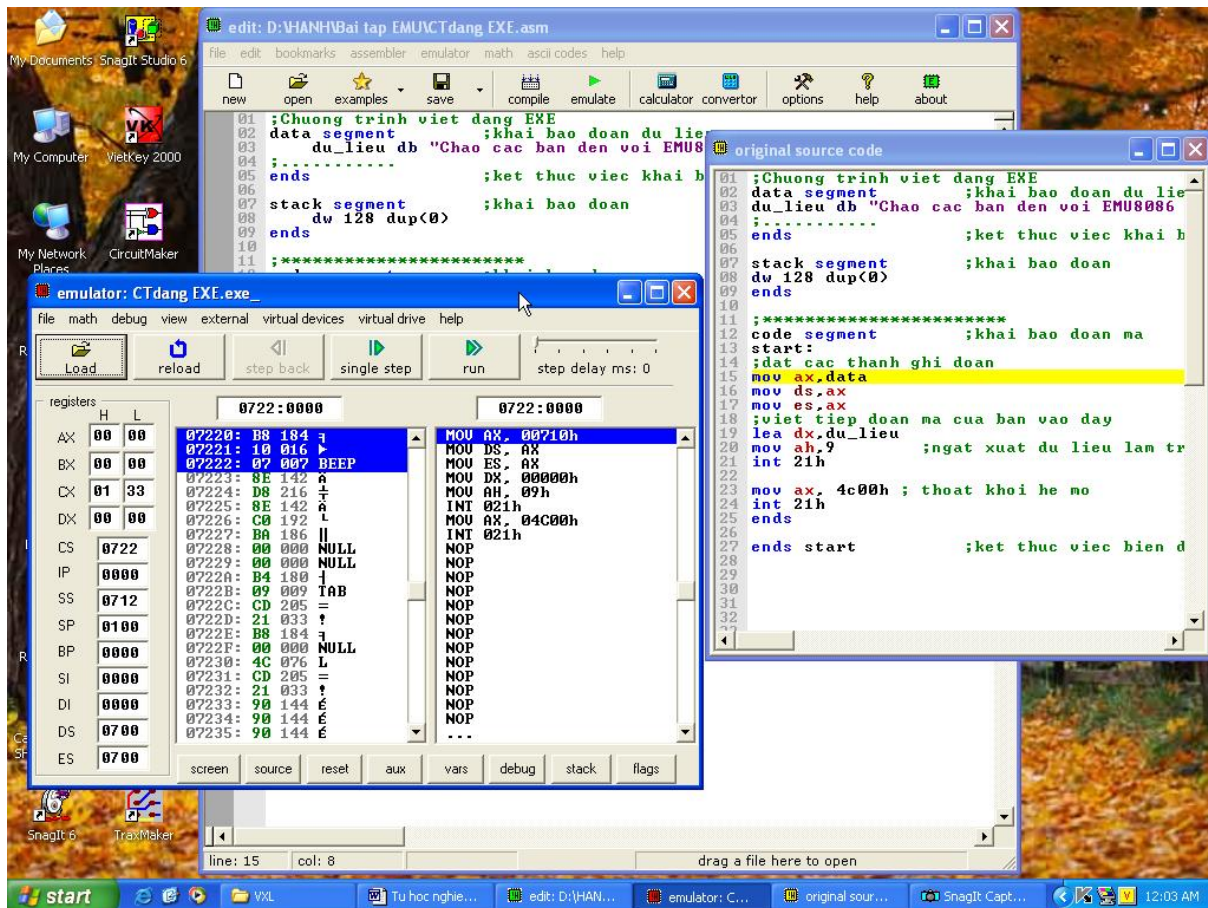
Máy đ- a ra thông báo lỗi ở dòng thứ 15 và lỗi dòng lệnh là: “move ax,data”. Dựa trên thông báo lỗi này có thể phán đoán dòng lệnh đ- ợc nhập sai mã lệnh. Để sửa lỗi này, nháy đúp vào thông báo lỗi đó, lúc này ch- ơng trình sẽ đánh dấu vết đen vào vị trí của dòng lệnh lỗi. Sau khi sửa xong phải tiến hành hợp dịch lại tr- ớc khi có thể chạy ch- ơng trình hay mô phỏng ch- ơng trình.

+ Đối với lỗi giải thuật: Với lỗi này máy tính không báo lỗi, hiện t- ợng khi mắc phải lỗi này là ch- ơng trình không chạy hoặc chạy không đúng yêu cầu mong muốn. Nguyên nhân của lỗi này th- ờng do sai giải thuật ch- ơng trình, sai

cấu trúc chương trình. Để tìm và sửa lỗi này cần chạy từng bước và quan sát kết quả thực hiện từng lệnh.

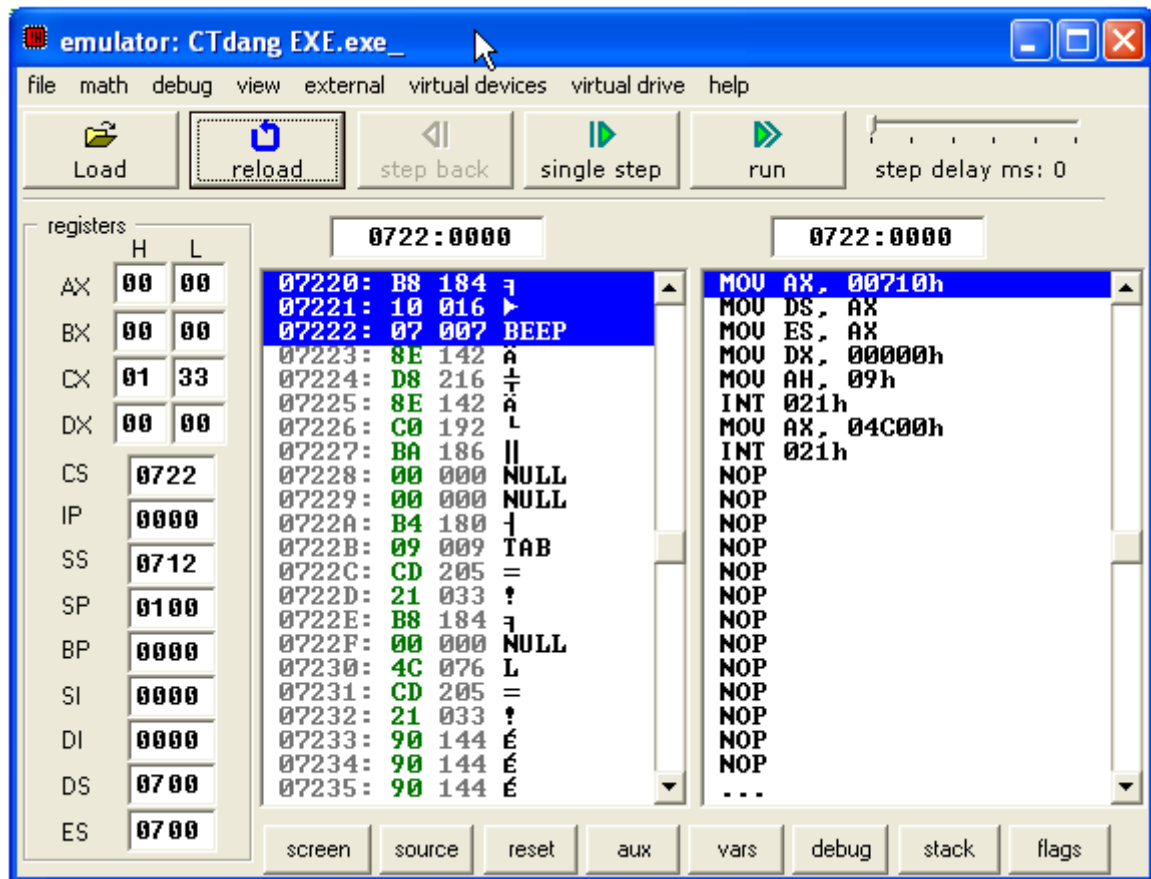
1.2.4. Mô phỏng và chạy chương trình

Để mô phỏng bạn có thể chọn ngay công cụ RUN của chương trình khi tiến hành hợp dịch mà không gặp lỗi hoặc cũng có thể chọn công cụ Emulate trên thanh công cụ. Sau khi lựa chọn chương trình đi ra các cửa sổ sau:



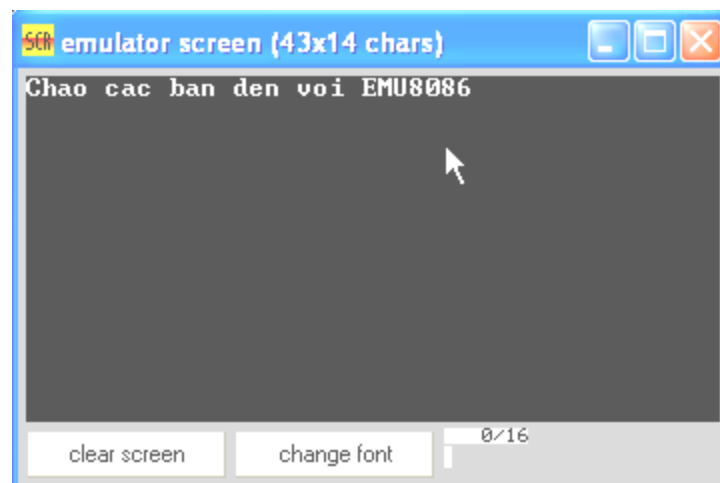
Cửa sổ Emulator cho phép mô phỏng chương trình và cửa sổ Original source code hiển thị mã và vị trí lệnh đang được thực hiện.

+ Giới thiệu về cửa sổ Emulator:



Cửa sổ này hiển thị các thanh ghi của vi xử lý nh- AX, BX, CX, DX, v.v. , địa chỉ các lệnh trong bộ nhớ, mã nhị phân t-ơng ứng của lệnh, các dòng lệnh và nhiều công cụ hỗ trợ cho việc mô phỏng khác.

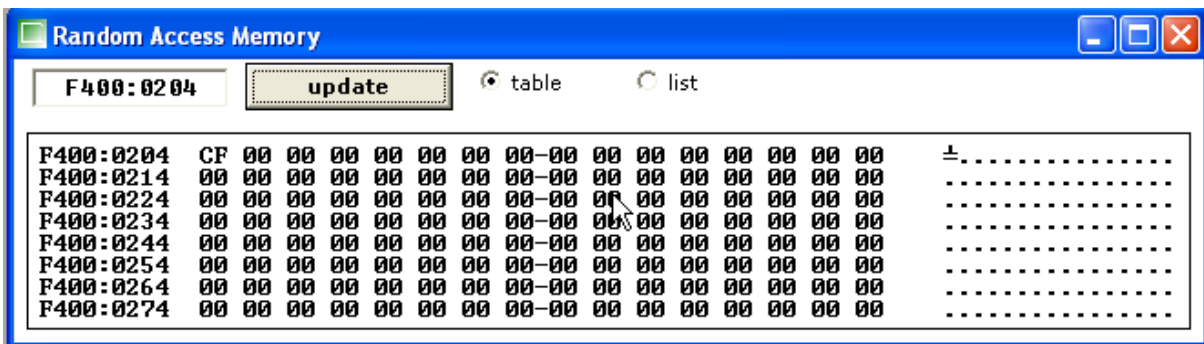
Chọn screen để quan sát màn hình.



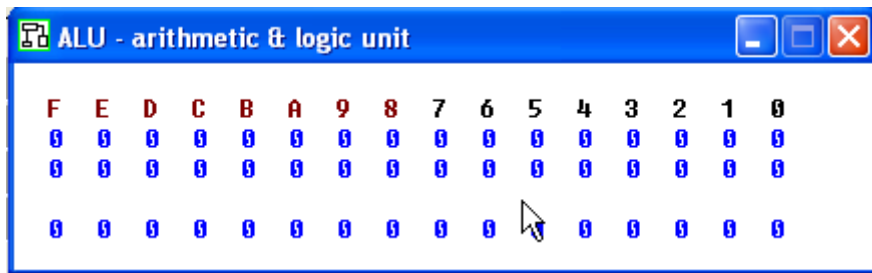
Chọn source để quan sát mã nguồn của ch-ơng trình. Chọn reset để khởi động lại quá trình mô phỏng.

Chọn aux để quan sát:

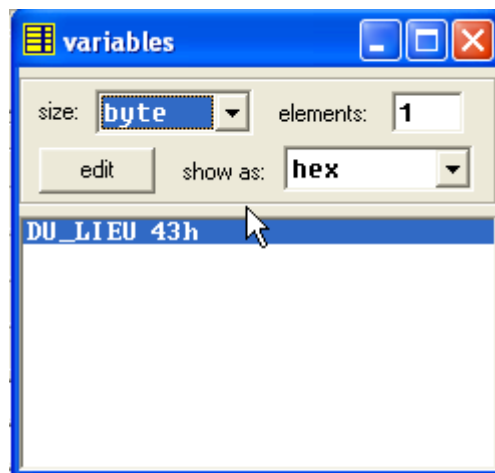
- Bộ nhớ (memory)



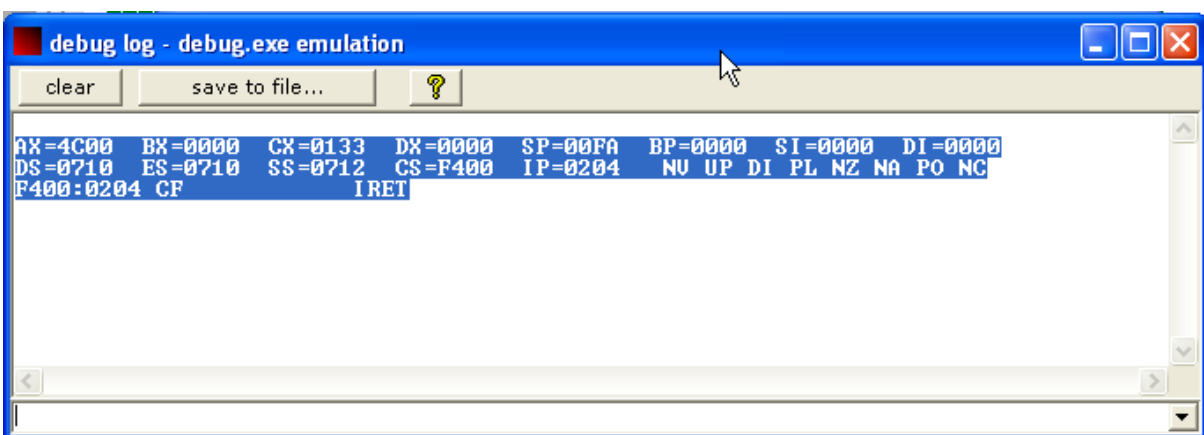
- Khối xử lý toán học (ALU)



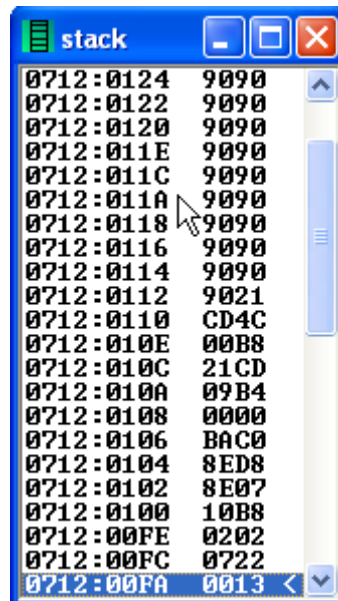
Chọn var để quan sát các biến, hằng...



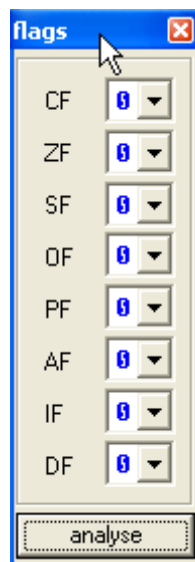
Chọn debug để quan sát các lệnh của chương trình dưới dạng trình gỡ rối debug



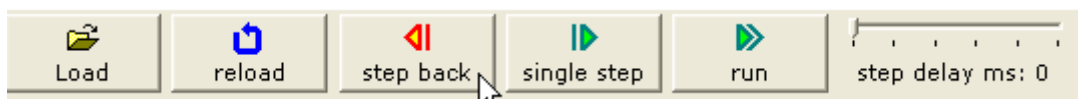
Chọn Stack để quan sát ngăn xếp



Chọn Flags để quan sát các cờ



Các công cụ hỗ trợ cho quá trình mô phỏng



- + Load: nạp một file chạy vào chương trình mô phỏng
- + Reload: nạp lại một chương trình hiện hành.
- + Step back: chạy lùi lại một bước
- + Single step: chạy tiến một bước
- + Run: chạy toàn bộ chương trình
- + Step delay ms: cho phép chọn tốc độ mô phỏng chương trình

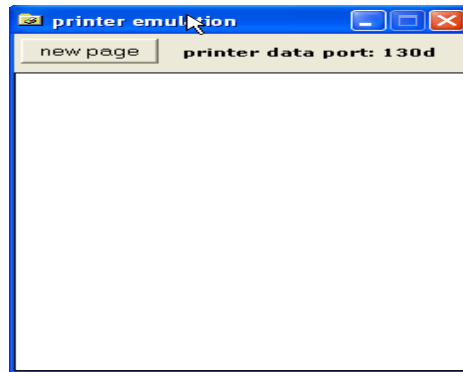
Các công cụ ảo trợ giúp mô phỏng: để nhanh chóng kiểm tra hoạt động của chương trình khi kết nối phần cứng chương trình hỗ trợ kết nối các thiết bị dưới dạng ảo. Mỗi thiết bị đều có một địa chỉ xác định tương ứng cho thiết bị ngoại vi bên ngoài và khi lập trình để xuất hay nhập dữ liệu thì chúng ta cũng làm tương tự như lập trình cho đối tượng thật.

Chọn Virtual Device:

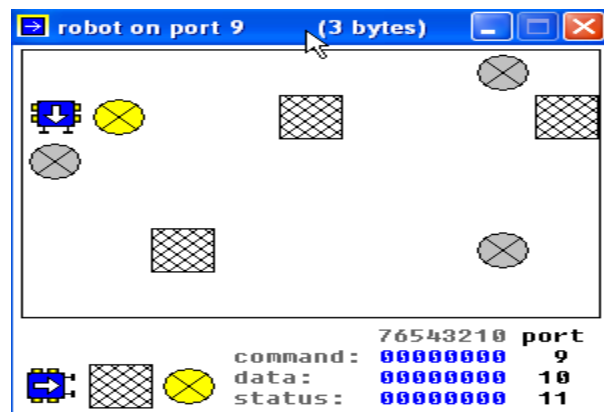
Led 7 đoạn: có địa chỉ là port 199



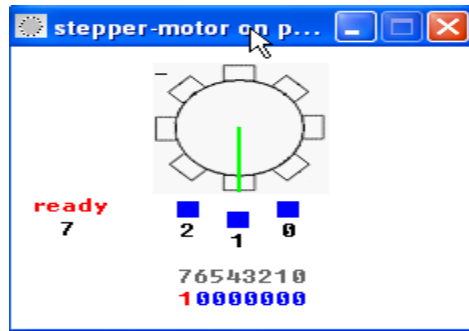
Máy in ảo có địa chỉ là port 130



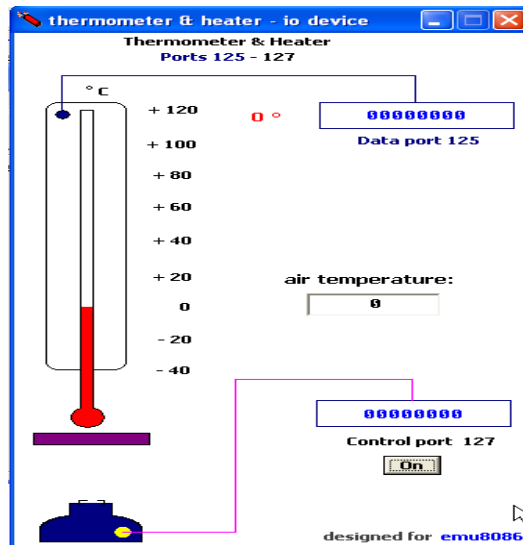
Mô phỏng robot có địa chỉ port là 9



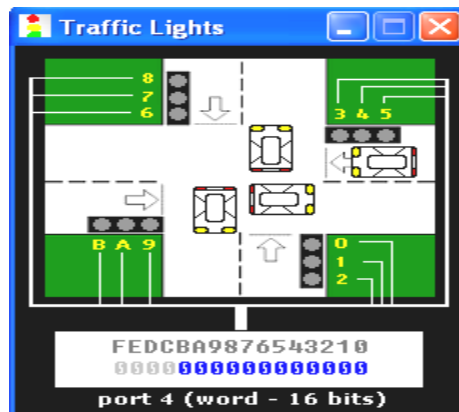
Mô phỏng động cơ bước có địa chỉ là port 7



Mô phỏng quá trình kiểm soát nhiệt độ có địa chỉ là port 125-127



Mô phỏng hoạt động của mạch giao thông có địa chỉ là port 4



1.3. Ví dụ và bài tập cơ bản lập trình cho vi xử lý 8086

Các bước viết chương trình cho vi xử lý trên EMU 8086

B-ÍC 1: Xác định, phân tích yêu cầu của bài toán từ đó vẽ giải thuật của chương trình cho bài toán đó.

B2: Khởi động ch- ơng trình EMU8086, mở bài mới, soạn thảo và ghi lại ch- ơng trình.

B3: Hợp dịch và sửa lỗi ch- ơng trình nếu có.

B4: Chạy ch- ơng trình hoặc thực hiện từng b- ớc để quan sát kết quả thực hiện lệnh, sửa lỗi giải thuật.

Ví dụ 1:

*Viết ch- ơng trình cho vxl hiển thị dòng thông báo " **chao mung cac ban den voi EMU 8086**" ra màn hình máy tính.*

B- ớc 1:

Phân tích: Để thực hiện đ- ọc yêu cầu trên thì bạn cần phải nhớ lại về hàm ngắt số 9 của ngắt INT 21h để hiển thị một chuỗi chữ ra màn hình.

Đối với một ch- ơng trình lớn, tiến hành vẽ l- u đồ giải thuật, các ch- ơng nhỏ thì có thể bỏ qua.

B- ớc2: Viết ch- ơng trình

```
name "hien thi" ;ten chuong trinh
;Chuong trinh viet dang EXE
.MODEL small ;khai bao mo hinh bo nho su dung cho chuong trinh
.DATA ;khai bao doan du lieu
du_lieu db "Chao mung cac ban den voi EMU8086 $"
.STACK 100 ;khai bao doan ngan xep
;*****
.CODE ;khai bao doan ma
Main Proc ;bat dau chuong trinh chinh
;xac lap gia tri cho thanh ghi doan du lieu
mov ax,data
mov ds,ax
lea dx,du_lieu ;nap du lieu vao dx
mov ah,9 ;ngat xuat du lieu lam trong ds ra man hinh( du lieu bat ;buoc
;ket thuc=)
```

```
int 21h
; ngat cho doi nhan phim
mov ah, 0
int 16h
mov ax, 4ch    ; thoat khoi chuong trinh
int 21h
Main endp
end Main      ;ket thuc chuong trinh
```

Sau khi soạn thảo ch- ơng trình, tiến hành dịch và chạy ch- ơng trình để kiểm tra kết quả.

Ví dụ 2:

Viết ch- ơng trình cho vi xử lý 80x86 thực hiện phép toán sau: $(5+10)-1=?$ Và hiển thị kết quả ra màn hình máy tính d- ới dạng số nhị phân.

B- ớc1:

Phân tích: Để thực hiện đ- ọc yêu cầu trên cần sử dụng các lệnh cộng, trừ và việc hiển thị kết quả ra màn hình d- ới dạng số nhị phân đòi hỏi phải hiển thị nội dung của từng bit sau đó thêm chữ b để thông báo đó là hệ nhị phân.

B- ớc2: Viết ch- ơng trình

```
name "cong va tru" ;ten chuong trinh
;chuong trinh viet theo khung dang COM
org 100h    ;khai bao dia chi bat dau chuong trinh

mov al, 5    ; nap gia tri cho al =5
mov bl, 10    ; nap gia tri cho al =10
; 5 + 10 = 15 (decimal) hoac hex=0fh hoac bin=00001111b
add bl, al    ;cong al voi bl
; 15 - 1 = 14 (decimal) hoac hex=0eh hoac bin=00001110b
sub bl, 1    ;tru bl di 1
;hien thi ket qua ra man hinh duoi dang so nhi phan
```

```
;day la so nhi phan 8 bit do do toi su dung vong lap 8lan
;dua 1 bit co trong bl ra man hinh su dung len xoay trai
;roi kiem tra co c, c=1 thi in 1, c=0 thi in 0. su dung
;ngat 2 cua int 21h de dua tung ky tu
mov cx, 8      ;so bit can dua ra
lap_in:
mov ah,2      ;ngat hien thi tung ky tu
clc           ;xoa co c
rol bl,1      ;xoay trai thanh ghi bl 1 vi tri
jc hien_thi1  ;kiem tra co c=1
mov dl,'0'    ;hien thi bit0
int 21h
jmp giam_cx
hien_thi1:
mov dl,'1'    ;hien thi bit1
int 21h
giam_cx:
loop lap_in
;in them chu B sau cac so nhi phan
mov dl, 'b'   ;nap ky tu can in
int 21h      ;ngat dua 1 ky tu ra man hinh
; ngat cho doi nhan phim:
mov ah, 0
int 16h
ret          ;ket thuc chuong trinh
```

Ví dụ 3:

*Viết chương trình cho vi xử lý 80x86 hiển thị dòng thông báo "**Hay nhập ký tự tu ban phím, nhập E để thoát**" ra màn hình máy tính và thực hiện chức năng nh- thông báo trên.*

B-ớc 1:

Phân tích: Để thực hiện đ-ợc yêu cầu trên thì bạn cần phải nhớ lại về hàm ngắt số 9 của ngắt INT 21h để hiển thị một chuỗi ký tự ra màn hình và hàm ngắt số 1 của INT 21h để nhận ký tự từ bàn phím, hàm ngắt số 2 của INT 21h để hiển thị từng ký tự.

B-ớc 2: Viết ch-ơng trình

```
name "NHAP PHIM" ;ten chuong trinh
;Chuong trinh viet dang EXE
.DATA ;khai bao doan du lieu
    du_lieu db "Hay nhap ky tu tu ban phim, nhap E de thoat$"
.STACK 100 ;khai bao doan ngan xep
.CODE ;khai bao doan ma
MAIN PROC
;dat cac thanh ghi doan
    mov ax,data ;muon AX de nap dia chi du lieu vao thanh ghi doan du lieu
    mov ds,ax
    lea dx,du_lieu ;nap du lieu vao dx
    mov ah,9 ;ngat xuat du lieu lam trong ds ra man hinh( du lieu bat buoc
;ket thuc=)
    int 21h
lap:mov ah,1 ;ngat nhan ky tu tu ban phim
    int 21h
    cmp al,'E' ;so sanh ky tu nhap vao trong Al voi E de thoat
    je thoat ;neu bang thi thoat
    mov dl,al ;chuyen sang dl de hien thi ra man hinh
    mov ah,2
    int 21h ;hien thi ky tu do ra man hinh
    jmp lap
thoat:
; ngat cho doi nhan phim
```

```
mov ah, 0
int 16h
mov ax, 4ch    ; thoát khỏi hệ thống
int 21h
MAIN ENDP
END MAIN      ; kết thúc việc hợp dịch chương trình
```

Ví dụ 4:

Viết chương trình cho VXL 8086 hiển thị thông báo: “ Đây là chương trình thực hiện cộng hai số nguyên dương có một chữ số được nhập từ bàn phím để hiển thị kết quả ra màn hình”, đồng thời thực hiện đọc chức năng của dòng thông báo trên.

B-ớc 1: Phân tích

- + Chương trình cần kiểm tra mã ASCII của các số nguyên dương từ 0-9. Nếu là các số nguyên từ 0-9 thì tiếp tục thực hiện, nếu sai yêu cầu nhập lại.
- + Chuyển đổi từ số nhị phân sang mã ASCII và ngược lại.
- + Hiển thị kết quả ra màn hình

B-ớc 2: Viết chương trình

```
name "Cong hai so nguyen duong co mot chu so"    ;ten chương trình
.model small    ;khai báo quy mô sử dụng chương trình
.stack 100      ;khai báo đoạn ngăn xếp
.data          ;khai báo đoạn dữ liệu
thong_bao:
db " Đây là chương trình thực hiện cộng hai số nguyên dương có một chữ số được nhập từ bàn phím để hiển thị kết quả ra màn hình $"
thong_bao1:
db " Hay nhập số a: $"
thong_bao2:
db "Hay nhập số b: $"
```



```
thong_bao3:
db "Ket qua a+b= $"
.code          ;khai bao doan ma
Main Proc      ;bat dau chuong trinh chinh

    mov ax,@data
    mov ds,ax
    lea dx,thong_bao
    mov ah,9
    int 21h
lap: lea dx,thong_bao1
    mov ah,9
    int 21h
    call nhap_so
    mov bl,al
    lea dx,thong_bao2
    mov ah,9
    int 21h
    call nhap_so
    mov bh,al
    lea dx,thong_bao3
    mov ah,9
    int 21h
    add bl,bh
    mov al,bl
    mov bl,10
    mov ah,0
    div bl
    add ah,30h
    mov bl,ah
    add al,30h
```

```
mov dl,al
call hien_thi
mov dl,bl
call hien_thi
jmp lap
nhap_so:
mov ah,0
int 16h
cmp al, '0'
jng nhap_so
cmp al, "9"
jg nhap_so
mov dl,al
call hien_thi
sub al,30h
ret
hien_thi:
mov ah,2
int 21h
ret
mov ah,4ch    ;ngat tro ve dos
int 21h
main endp    ;ket thuc chuong trinh chinh
end main     ;ket thuc chuong trinh
```

Bài tập:

Bài tập 1: Viết chương trình để hiển thị tên của bạn trên màn hình

Bài tập 2: Cho vi xử lý thực hiện phép toán sau $(15+255+75) * 25$ và in kết quả ra màn hình dạng số nhị phân

Bài tập 3: Viết chương trình cho vi xử lý thực hiện phép toán sau:

$(15/5)+3*4$ và in kết quả ra màn hình dưới dạng số hexa

Bài tập 4: Viết chương trình thực hiện nhân hai số tự nhiên từ 0-9 bất kỳ nhập vào từ bàn phím. Hiển thị tích lên màn hình.

Bài tập 5: Viết chương trình thực hiện cộng hai số tự nhiên từ 0-99 bất kỳ nhập vào từ bàn phím. Hiển thị tổng lên màn hình.