

به نام خدا

پروژه ی معماری: شبیه ساز کامپیوتر مانو

حدیث غفوری ۹۸۲۵۴۱۳

زبان های استفاده شده: html,css,js

روش های اجرای برنامه

۱: کلیک بر فایل index.html

۲: مشاهده ی سایت hadis98.github.io/Mano-simulator/

حالت کلی برنامه

The screenshot shows the Mano Simulator web application. It features a 'Code Editor' on the left, a central 'Instructions Detail' table, and a 'Memory Table' on the right. Below the instructions table are buttons for 'Fetch', 'Decode', and 'Execute', along with a status box showing 'instruction executed'. An 'Assemble' button is located at the bottom left.

MANO SIMULATOR

Instructions Detail

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values							
T0. AR ← PC							
T1. IR ← M[AR], PC ← PC+1							
T2. AR ← IR[0..11]							
T3							
T4							
T5							
T6							
After Execution							

Memory Table

Decimal Address	Hex Address	Contents
0	0x000	
1	0x001	
2	0x002	
3	0x003	
4	0x004	
5	0x005	
6	0x006	
7	0x007	
8	0x008	
9	0x009	
10	0x00A	
11	0x00B	
12	0x00C	
13	0x00D	
14	0x00E	
15	0x00F	
16	0x010	
17	0x011	
18	0x012	
19	0x013	

برای استفاده از این برنامه

1: ابتدا کد مورد نظر را به صورت صحیح در کد ادیتور وارد کنید (مطابق با فرمت مثال)

The screenshot shows the Mano Simulator interface. On the left, the Code Editor contains the following assembly code:

```
ORG 100
LDA SUB
CMA
INC
ADD MIN
STA DIF
HLT
MIN, DEC 83
SUB, DEC -23
DIF, HEX 0
END
```

The Instructions Detail table shows the execution of the HLT instruction:

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values	0x3108	0x6A	0x53	0x105	0x108	0x6A	0x0
T0. AR ← PC	0x3108	0x6A	0x53	0x105	0x105	0x7001	0x0
T1. IR ← M[AR], PC ← PC+1	0x7001	0x6A	0x53	0x106	0x105	0x7001	0x0
T2. AR ← IR[0..11]	0x7001	0x6A	0x53	0x106	0x1	0x0	0x0
T3. S ← 0							
T4.							
T5.							
T6.							
After Execution							

The Memory Table on the right shows the state of memory:

Decimal Address	Hex Address	Contents
256	0x100	2107
257	0x101	7200
258	0x102	7020
259	0x103	1106
260	0x104	3108
261	0x105	7001
262	0x106	0053
263	0x107	FFE9
264	0x108	006A
265	0x109	
266	0x10A	
267	0x10B	
268	0x10C	
269	0x10D	
270	0x10E	
271	0x10F	
272	0x110	
273	0x111	
274	0x112	
275	0x113	

The status bar at the bottom indicates "instruction HLT executed".

توجه: پس از آخرین دستور در ادیتور نباید enter وارد کنید در غیر این صورت خطای آخرین دستور را دریافت میکنید. مانند مثال زیر:

The screenshot shows the Mano Simulator interface with an error message displayed in the center: "an invalid instruction in address 010F of program". The Code Editor contains the following assembly code:

```
ORG 100
LDA AL
ADD BL
STA CL
CLA
CIL
ADD AH
ADD BH
STA CH
HLT
AL, DEC 12
AH, DEC 32
BL, DEC 40
BH, DEC 02
CL, DEC 0
CH, DEC 0
END
```

The Instructions Detail table shows the execution of the HLT instruction:

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values							
T0. AR ← PC							
T1. IR ← M[AR], PC ← PC+1							
T2. AR ← IR[0..11]							
T3							
T4							
T5							
T6							
After Execution							

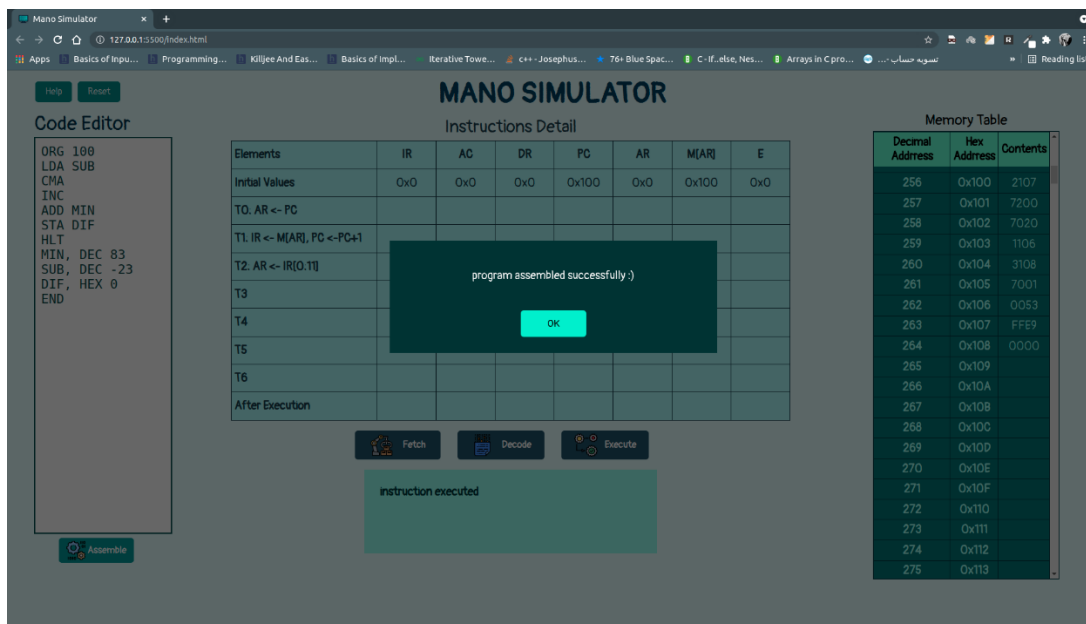
The Memory Table on the right shows the state of memory:

Decimal Address	Hex Address	Contents
0	0x000	
1	0x001	
2	0x002	
3	0x003	
4	0x004	
5	0x005	
6	0x006	
7	0x007	
8	0x008	
9	0x009	
10	0x00A	
11	0x00B	
12	0x00C	
13	0x00D	
14	0x00E	
15	0x00F	
16	0x010	
17	0x011	
18	0x012	
19	0x013	

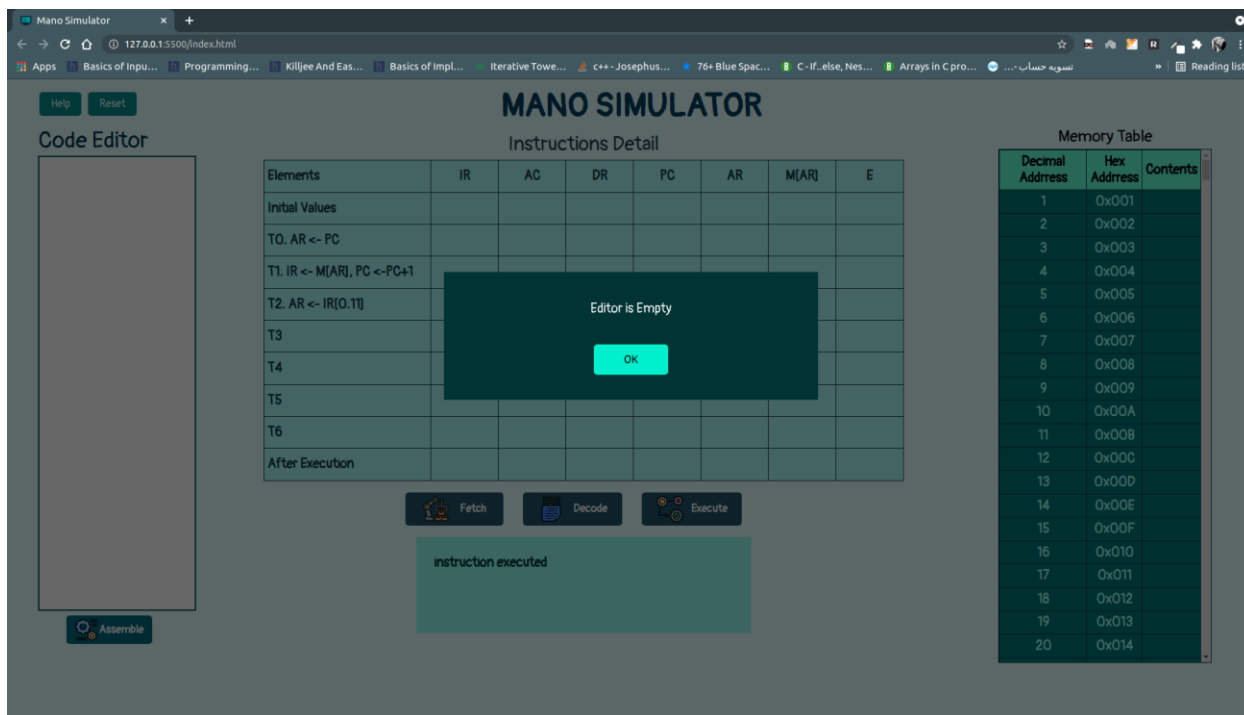
The status bar at the bottom indicates "instruction executed".

2: سپس روی دکمه ی اسمبل کلیک کنید.

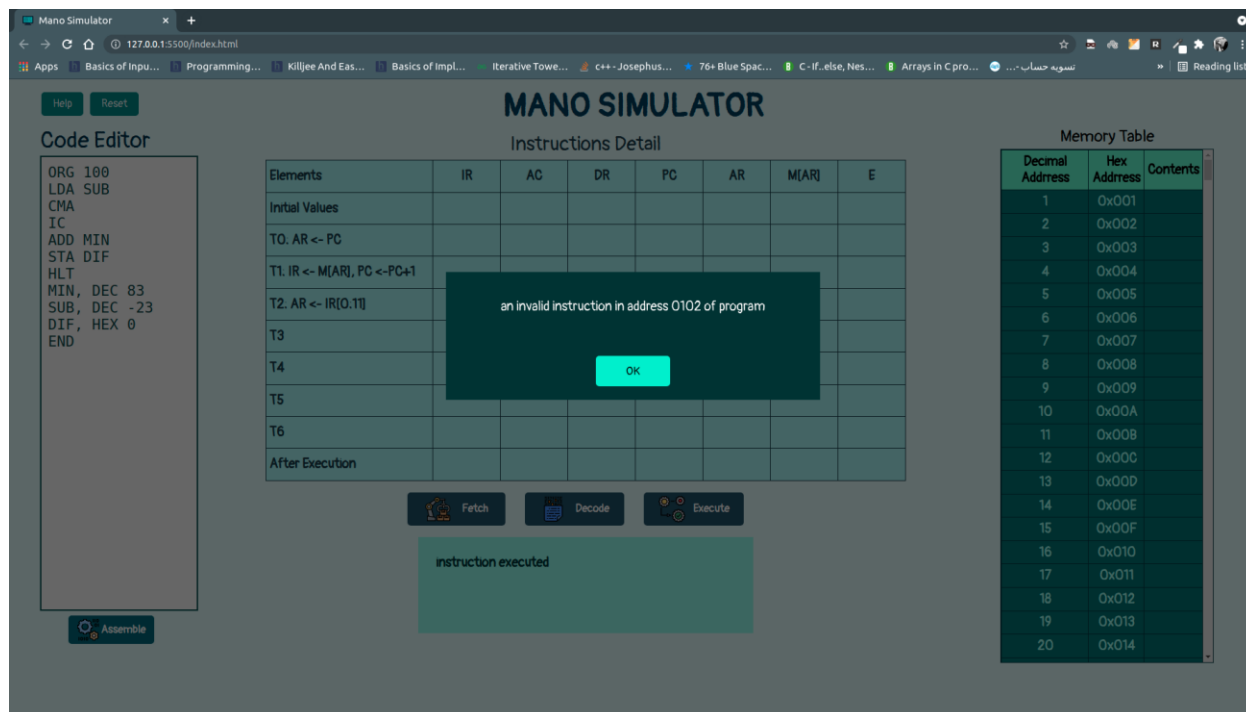
با کلیک کردن روی این دکمه پنجره ای ظاهر میشود که میگوید عملیات اسمبل موفقیت آمیز بود.



دقت کنید اگر روی دکمه اسمبل کلیک کنید درحالی که ادیتور خالی باشد ارور زیر را دریافت میکنید.



نکته ی مهم دیگر: اگر دستوری را به صورت اشتباه وارد کنید یا یک متغیر که در انتهای برنامه تعریف کرده اید را اشتباه وارد کنید اروری مانند مثال زیر دریافت میکنید.



در صورت موفق بودن عملیات ادرس ها و محتوای آنها در جدول سمت راست که مربوط به حافظه است نمایش داده میشود.



3: در این مرحله برای اجرای دستور ها باید برای هر دستور ابتدا روی دکمه ی fetch سپس دکمه ی decode و سپس execute کلیک کنید. دقت کنید دکمه ها در صورتی فعال میشوند که شما این ترتیب را رعایت کنید.

با کلیک بر دکمه ی fetch و decode و execute جدول وسط که مربوط به دستور ها است تغییر میکند و همچنین مطابق با دکمه ی فشار داده شده حالت اجرایی در پایین صفحه نمایش داده میشود مطابق با تصویر بالا.

همچنین برای سهولت ، هر دستور که اجرا میشود در کادر نشان داده میشود مانند تصویر بالا.

هر زمان که جدول مموری ابدیت شود در کادر پایین برنامه نمایش داده میشود همچنین مقدار ابدیت شده در جدول با رنگی متفاوت در جدول قرار میگیرد.

The screenshot shows the MANO Simulator interface. On the left is the Code Editor with assembly code. In the center is the Instructions Detail table showing the state of various registers and the current instruction. On the right is the Memory Table showing memory addresses and their contents. A red box highlights the instruction 'STA' being executed, and another red box highlights the memory address 0108 where the content 6A is being stored.

Code Editor

```

ORG 100
LDA SUB
CMA
INC
ADD MIN
STA DIF
HLT
MIN, DEC 83
SUB, DEC -23
DIF, HEX 0
END
  
```

Instructions Detail

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values	0x1106	0x6A	0x53	0x104	0x106	0x0053	0x0
T0. AR ← PC	0x1106	0x6A	0x53	0x104	0x104	0x3108	0x0
T1. IR ← M[AR], PC ← PC+1	0x3108	0x6A	0x53	0x105	0x104	0x3108	0x0
T2. AR ← IR[0:11]	0x3108	0x6A	0x53	0x105	0x108	0x0000	0x0
T3.							
T4. M[AR] ← AC, SC ← 0	0x3108	0x6A	0x53	0x105	0x108	0x6A	0x0
T5.							
T6.							
After Execution	0x3108	0x6A	0x53	0x105	0x108	0x6A	0x0

Memory Table

Decimal Address	Hex Address	Contents
256	0x100	2107
257	0x101	7200
258	0x102	7020
259	0x103	1106
260	0x104	3108
261	0x105	7001
262	0x106	0053
263	0x107	FFB9
264	0x108	006A
265	0x109	
266	0x10A	
267	0x10B	
268	0x10C	
269	0x10D	
270	0x10E	
271	0x10F	
272	0x110	
273	0x111	
274	0x112	
275	0x113	

Execution Log: instruction STA executed
content 6A has been added to address 0108 of memory

در انتهای برنامه کادری نمایش داده میشود که برنامه به آخر رسید.

دقت کنید برای استفاده ی مجدد از برنامه حتما باید بر دکمه ی reset کلیک کنید.

Mano Simulator

127.0.0.1:5500/index.html

Help Reset

MANO SIMULATOR

Code Editor

```

ORG 100
LDA SUB
CMA
INC
ADD MIN
STA DIF
HLT
MIN, DEC 83
SUB, DEC -23
DIF, HEX 0
END

```

Instructions Detail

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values	0x3108	0x6A	0x53	0x105	0x108	0x6A	0x0
T0. AR ← PC	0x3108	0x6A	0x53	0x105	0x105	0x7001	0x0
T1. IR ← M[AR], PC ← PC+1	0x7001	0x6A	0x53	0x106	0x105	0x7001	0x0
T2. AR ← IR[0..11]							0x0
T3. S ← 0							
T4.							
T5.							
T6.							
After Execution							

End of Program:
to test another example click on RESET button

Fetch Decode Execute

instruction HLT executed

Assemble

executed

Memory Table

Decimal Address	Hex Address	Contents
256	0x100	2107
257	0x101	7200
258	0x102	7020
259	0x103	1106
260	0x104	3108
261	0x105	7001
262	0x106	0053
263	0x107	FFE9
264	0x108	006A
265	0x109	
266	0x10A	
267	0x10B	
268	0x10C	
269	0x10D	
270	0x10E	
271	0x10F	
272	0x110	
273	0x111	
274	0x112	
275	0x113	

در انتهای برنامه همه ی دکمه ها قفل میشوند تا زمانی که دکمه ی reset زده شود.

Mano Simulator

127.0.0.1:5500/index.html

Help Reset

MANO SIMULATOR

Code Editor

```

ORG 100
LDA SUB
CMA
INC
ADD MIN
STA DIF
HLT
MIN, DEC 83
SUB, DEC -23
DIF, HEX 0
END

```

Instructions Detail

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values	0x3108	0x6A	0x53	0x105	0x108	0x6A	0x0
T0. AR ← PC	0x3108	0x6A	0x53	0x105	0x105	0x7001	0x0
T1. IR ← M[AR], PC ← PC+1	0x7001	0x6A	0x53	0x106	0x105	0x7001	0x0
T2. AR ← IR[0..11]	0x7001	0x6A	0x53	0x106	0x1	0x0	0x0
T3. S ← 0							
T4.							
T5.							
T6.							
After Execution							

Fetch Decode Execute

instruction HLT executed

Assemble

executed

Memory Table

Decimal Address	Hex Address	Contents
256	0x100	2107
257	0x101	7200
258	0x102	7020
259	0x103	1106
260	0x104	3108
261	0x105	7001
262	0x106	0053
263	0x107	FFE9
264	0x108	006A
265	0x109	
266	0x10A	
267	0x10B	
268	0x10C	
269	0x10D	
270	0x10E	
271	0x10F	
272	0x110	
273	0x111	
274	0x112	
275	0x113	

با کلیک بر دکمه ی help اطلاعاتی درباره ی برنامه نمایش داده میشود.

MANO SIMULATOR

About

this program is a simulator for basic MANO computer

It has a memory table that shows address and content of memory

It has a instruction table that shows detail of calculations

How to use this program?

- 1.place your code in editor
- 2.click on Assemble button
- 3.for every instruction, click on fetch button then decode button then execute button

NOTE: you should keep the order of clicking on buttons otherwise it wont work properly

NOTE: to use program again, click on reset button

created By HADIS GHAFOURI

view the prodeess of coding in my github

Close

MANO SIMULATOR

Instructions Detail

Elements	IR	AC	DR	PC	AR	M[AR]	E
Initial Values							
T0: AR <- PC							
T1: IR <- M[AR], PC <- PC+1							
T2: AR <- IR[0..11]							
T3							
T4							
T5							
T6							
After Execution							

Fetch
 Decode
 Execute

instruction executed

Memory Table

Decimal Address	Hex Address	Contents
1	0x001	
2	0x002	
3	0x003	
4	0x004	
5	0x005	
6	0x006	
7	0x007	
8	0x008	
9	0x009	
10	0x00A	
11	0x00B	
12	0x00C	
13	0x00D	
14	0x00E	
15	0x00F	
16	0x010	
17	0x011	
18	0x012	
19	0x013	
20	0x014	
21	0x015	
22	0x016	
23	0x017	

با کلیک بر دکمه ی گیت هاب در منوی help میتوانید روند کدزدن را در گیت هاب مشاهده کنید.

همچنین میتوانید مثال های دیگری (مانند شیفت ۴بیتی و ضرب ۲عدد و تابع OR و تفریق ۲ عدد و جمع دو عدد بزرگ 32 بیتی) از برنامه را در فولدر project_photos مشاهده کنید.


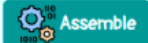


Code Editor

```

ORG 100
LDA AL
ADD BL
STA CL
CLA
CIL
ADD AH
ADD BH
STA CH
HLT
AL, DEC 12
AH, DEC 32
BL, DEC 40
BH, DEC 02
CL, DEC 0
CH, DEC 0
END

```

Assemble

Code Editor	Code Editor	Code Editor	Code Editor
<pre>ORG 100 LDA SUB CMA INC ADD MIN STA DIF HLT MIN, DEC 83 SUB, DEC -23 DIF, HEX 0 END</pre>	<pre>ORG 100 LDA X BSA SH4 STA X LDA Y BSA SH4 STA Y HLT X, HEX 1234 Y, HEX 4321 SH4, HEX 0 CIL CIL CIL CIL AND MSK BUN SH4 I MSK, HEX FFF0 END</pre>	<pre>ORG 100 LDA X BSA OR HEX 140 STA Y HLT X, HEX 150 Y, HEX 0 OR, HEX 0 CMA STA TMP LDA OR I CMA AND TMP CMA ISZ OR BUN OR I TMP, HEX 0 END</pre>	<pre>ORG 100 LOP, CLE LDA Y CIR STA Y SZE BUN ONE BUN ZRO ONE, LDA X ADD P STA P CLE ZRO, LDA X CIL STA X ISZ CTR BUN LOP HLT CTR, DEC -8 X, HEX F Y, HEX B P, HEX 0 END</pre>
			

در انتهای فایل app.js این کدها به صورت کامنت برای دسترسی آسان تر قرار داده شده اند.