Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2023), B.Sc. in CSE (EVE)

Children Learning Alphabet and Digit in Assembly Language

Course Title:Microprocessor and Microcontroller Lab
Course Code: CSE 304
Section: 222EA

Students Details

Name	ID
Sadia Islam Sraban	- 222015025
Samayun mia Chowdhury-	- 222015031

Course Teacher's Name: _ Mahmuda Rahman

[For teachers use only: Don't write anything inside this box]
olar

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

Chapter 1

Introduction

Teaching children about the alphabet and digits using assembly language can be an interesting approach, though it might be a bit unconventional. Assembly language is a low-level programming language that's more closely tied to the hardware of a computer system. It might not be the most intuitive way to teach kids, but it can offer a unique perspective on how computers understand and process information.

1.1 Overview

By integrating these motivational elements into assembly language-based learning tools, you can create an engaging and stimulating environment that encourages children to explore, experiment, and master the alphabet and digits while having fun with programming concepts.

1.2 Motivation

In summary, choosing a project that involves teaching children the alphabet and digits using assembly language offers a unique opportunity to blend technology, education, and creativity. 1.Low-Level Understanding. 2.Innovation in Education. 3.Creative Expression. 4.Alphabet and Digit Mastery.

1.3 Problem Definition

.

1.3.1 Problem Statement

1.Target Audience: Children between a certain age range (3-8 years old) who are beginning to learn letters and numbers.

2.Design the program to allow for user interaction, such as input via keyboard or mouse, to progress through the alphabet and digits.

1.3.2 Complex Engineering Problem

Balance the complexity of the program with the target audience's cognitive abilities to ensure it's challenging yet comprehensible. Consider the limitations of Assembly Language in terms of memory, processing power, and graphical capabilities while designing the program.

Table 1.1: Summary of the attributes touched by the mentioned projects

Name of the P Attributess	**Explain how to address**
P1: Depth of knowledge required	Basic assembly language concepts, Alphabet and digit representation.
P2: Range of conflicting requirements	-Ideally, children grasp the underlying logic of instructions, data types, and memory manipulation. This requires a more technical approach.
P3: Depth of analysis required	-Analyze the effectiveness of different visual representations and explanations in promoting understanding.
P4: Familiarity of issues	-Children might already have some understanding of letters, digits, and even basic computer concepts. This prior knowledge can serve as a foundation for building upon.
P5: Extent of applicable codes	-Displaying letters and digits,Matching games and exercises,
P6: Extent of stakeholder involvement and conflicting requirements	-Various stakeholders are involved, each with potential conflicting requirements. Balancing these requirements is crucial for creating a successful and impactful learning experience.
P7: Interdependence	They influence and interweave with each other to create a holistic learning experience.

1.4 Design Goals/Objectives

1.Create a program that captivates children's attention.

2. Incorporate visual representations to aid understanding.

3.Design interactive elements that allow children to participate actively in the learning process.

4.Ensure the program is accessible and easy to use for children of various ages and skill levels.

1.5 Application

Fundamental Understanding of Computing: Introducing assembly language at an early stage helps children grasp the foundational concepts of how computers interpret and display characters.

Critical Thinking and Problem-Solving: Learning assembly language encourages critical thinking and problem-solving skills.

Introduction to Programming Concepts: Assembly language exposes children to programming concepts such as data representation, memory management, and basic instructions, laying a strong groundwork for future learning in computer science.

Visualizing Abstract Concepts: Assembly language can be abstract, but teaching the alphabet and digits in this manner can involve creating visual displays or animations to illustrate how characters are represented in binary, making abstract concepts more tangible.

Creating Educational Tools: Developing educational software or games that use assembly language to teach the alphabet and digits can serve as unique learning tools.

Chapter 2

Design/Development/Implementation of the Project

2.1 Introduction

In assembly language, teaching children the alphabet and digits involves breaking down complex concepts into simpler instructions that a computer can understand. In real assembly language programming, the exact instructions will vary depending on the processor architecture and hardware used. This code focuses on the conceptual steps of displaying instructions, looping through the alphabet and digits, and displaying them on a theoretical screen or output device.

2.2 Project Details

1. Target age group: Knowing the age range of the children will help me tailor the difficulty level and focus of the learning system.

2. Learning objectives: What do you want children to learn from this system? Is it simply alphabet and digit recognition, or do you want them to delve deeper into basic programming concepts?

3. Technical limitations: Are there any specific assembly language architectures or limitations you need to consider?

I'm excited to help you create a meaningful and impactful learning experience for children!

2.3 Project Tools

1.Assembly language

2.Emu 8086 compiler used to built the project

3.Data bus used

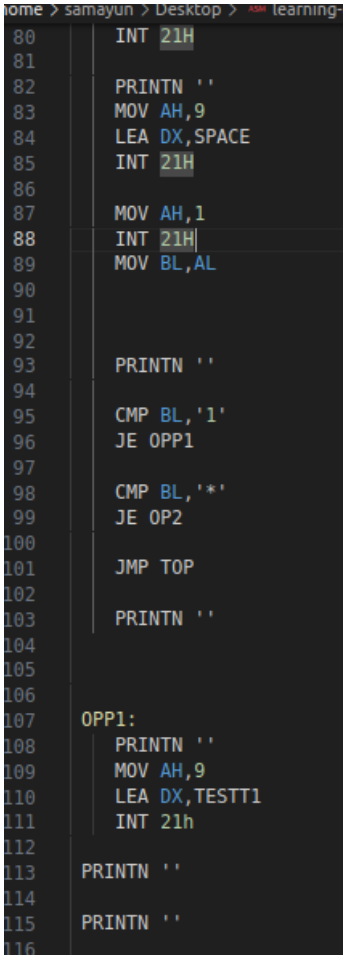
2.4 Implementation

learning-app.asm

home > samayun > Desktop > learning-app.asm

```
.MODEL SMALL
.STACK 100H
.DATA

TESTT DB "                  1.STUDY OR *.Exit$"
TESTT1 DB "                     STUDY",0dh,0ah
       DB "                Input Alphabet or Digit$",0dh,0ah
 FOR DB " for $"

Z1 DB "ZERO.$"
Z2 DB "ONE.$"
Z3 DB "TWO.$"
Z4 DB "THREE.$"
Z5 DB "FOUR.$"
Z6 DB "FIVE.$"
Z7 DB "SIX.$"
Z8 DB "SEVEN.$"
Z9 DB "EIGHT.$"
Z10 DB "NINE.$"

MSG db "                      ASSEMBLY PROJECT", 0dh,0ah
    db "                ==== Children Learning App ====", 0dh,0ah
  db "                    ====================", 0dh,0ah
  db "                    Press any key to start...", 0dh,0ah
  db "                                           $", 0dh,0ah


a    db "Apple.$"
b    db "Ball.$"
c    db "Cat.$"
d    db "Dog.$"
e    db "Egg.$"
f    db "Fan.$"
g    db "Goat.$"
h    db "Hen.$"
i    db "Ink.$"
j    db "Jackfruit.$"
k    db "Kite.$"
l    db "Lion.$"
```

```
s    db "Sheep.$"
t    db "Tiger.$"
u    db "Umbrella.$"
v    db "Violin.$"
w    db "Window.$"
x    db "X-ray.$"
y    db "Yolk.$"
z    db "Ziraf.$"

TNX DB "                    ===o THANK YOU o===$"
INV DB "                          INVALID INPUT.$"
SPACE DB "                         $"

.CODE

MAIN PROC

    MOV AX,@DATA
    MOV DS,AX

    INCLUDE 'EMU8086.INC'

    MOV AH,9
    LEA DX,MSG
    INT 21H

    MOV AH,1
    INT 21H

    TOP:
    PRINTN ''

    MOV AH,9
    LEA DX,TESTT
    INT 21H
```


ome > samayun > Desktop > learning-
INT 21H
PRINTN ''
MOV AH,9
LEA DX,SPACE
INT 21H
MOV AH,1
INT 21H
MOV BL,AL
PRINTN ''
CMP BL,'1'
JE OPP1
CMP BL,'*'
JE OP2
JMP TOP
PRINTN ''
OPP1:
PRINTN ''
MOV AH,9
LEA DX,TESTT1
INT 21h
PRINTN ''
PRINTN ''

```
115  PRINTN ''
116
117   OP1:
118  PRINTN ''
119
120  MOV AH,9
121  LEA DX,SPACE
122  INT 21H
123
124
125  MOV AH,1
126  INT 21H
127  MOV BL,AL
128
129
130   CMP BL,'*'
131  JE OP2
132
133  CMP BL,58
134  JL NUMBER
135
136  MOV AH,9
137  LEA DX,FOR
138  INT 21H
139
140
141  CMP  BL,'A'
142  JE C1
143  CMP BL,'a'
144  JE C1
145
146  CMP  BL,'B'
147  JE C2
148  CMP BL,'b'
149  JE C2
```

```
146  CMP  BL,'B'
147  JE C2
148  CMP BL,'b'
149  JE C2
150
151  CMP  BL,'C'
152  JE C3
153  CMP BL,'c'
154  JE C3
155
156  CMP  BL,'D'
157  JE C4
158  CMP BL,'d'
159  JE C4
160
161  CMP  BL,'E'
162  JE C5
163  CMP BL,'e'
164  JE C5
165
166  CMP  BL,'F'
167  JE C6
168  CMP BL,'f'
169  JE C6
170
171  CMP  BL,'G'
172  JE C7
173  CMP BL,'g'
174  JE C7
175
176  CMP  BL,'H'
177  JE C8
178  CMP BL,'h'
179  JE C8
```

```
180
181 CMP  BL,'I'
182 JE C9
183 CMP BL,'i'
184 JE C9
185
186 CMP  BL,'J'
187 JE C10
188 CMP BL,'j'
189 JE C10
190
191 CMP  BL,'K'
192 JE C11
193 CMP BL,'k'
194 JE C11
195
196 CMP  BL,'L'
197 JE C12
198 CMP BL,'l'
199 JE C12
200
201 CMP  BL,'M'
202 JE C13
203 CMP BL,'m'
204 JE C13
205
206 CMP  BL,'N'
207 JE C14
208 CMP BL,'n'
209 JE C14
210
211 CMP  BL,'O'
212 JE C15
213 CMP BL,'o'
214 JE C15
```

```
216 CMP  BL,'P'
217 JE C16
218 CMP BL,'p'
219 JE C16
220
221 CMP  BL,'Q'
222 JE C17
223 CMP BL,'q'
224 JE C17
225
226 CMP  BL,'R'
227 JE C18
228 CMP BL,'r'
229 JE C18
230
231 CMP  BL,'S'
232 JE C19
233 CMP BL,'s'
234 JE C19
235
236 CMP  BL,'T'
237 JE C20
238 CMP BL,'t'
239 JE C20
240
241 CMP  BL,'U'
242 JE C21
243 CMP BL,'u'
244 JE C21
245
246 CMP  BL,'V'
247 JE C22
248 CMP BL,'v'
249 JE C22
```

```
CMP  BL,'W'
JE C23
CMP BL,'w'
JE C23

CMP  BL,'X'
JE C24
CMP BL,'x'
JE C24

CMP  BL,'Y'
JE C25
CMP BL,'y'
JE C25

CMP  BL,'Z'
JE C26
CMP BL,'z'
JE C26

 NUMBER:

 MOV AH,2
 MOV DL,8
 INT 21H

  CMP  BL,'0'
JE C27

 CMP  BL,'1'
JE C28

CMP BL,'2'
JE C29

 CMP  BL,'3'
JE C30
```

```
 CMP  BL,'1'
JE C28

CMP BL,'2'
JE C29

 CMP  BL,'3'
JE C30

CMP BL,'4'
JE C31

 CMP  BL,'5'
JE C32

CMP BL,'6'
JE C33


 CMP  BL,'7'
JE C34

CMP BL,'8'
JE C35

 CMP  BL,'9'
JE C36
 JMP OP1

 MOV AH,9
 LEA DX,INV
 INT 21H
   JMP OP1

```

```
314     C1:
315     MOV AH,9
316    LEA DX,a
317     INT 21H
318     JMP OP1
319
320     C2:
321     MOV AH,9
322    LEA DX,b
323     INT 21H
324       JMP OP1
325
326     C3:
327     MOV AH,9
328    LEA DX,c
329     INT 21H
330       JMP OP1
331
332     C4:
333     MOV AH,9
334    LEA DX,d
335     INT 21H
336     JMP OP1
337
338     C5:
339     MOV AH,9
340    LEA DX,e
341     INT 21H
342       JMP OP1
343
344     C6:
345     MOV AH,9
346    LEA DX,f
347     INT 21H
348       JMP OP1
349
```

```
344     C6:
345     MOV AH,9
346    LEA DX,f
347     INT 21H
348       JMP OP1
349
350     C7:
351     MOV AH,9
352    LEA DX,g
353     INT 21H
354      JMP OP1
355
356     C8:
357     MOV AH,9
358    LEA DX,h
359     INT 21H
360      JMP OP1
361
362     C9:
363     MOV AH,9
364    LEA DX,i
365     INT 21H
366      JMP OP1
367
368     C10:
369     MOV AH,9
370
371     lEA DX,j
372     INT 21H
373       JMP OP1
374
375     C11:
376     MOV AH,9
Run Testcases  ⊗ 0 ⚠ 0   0
```

```
 C11:
 MOV AH,9
LEA DX,k
 INT 21H
   JMP OP1

 C12:
 MOV AH,9
LEA DX,l
 INT 21H
   JMP OP1

 C13:
 MOV AH,9
LEA DX,m
 INT 21H
  JMP OP1

 C14:
 MOV AH,9
LEA DX,n
 INT 21H
  JMP OP1

 C15:
 MOV AH,9
LEA DX,o
 INT 21H
  JMP OP1

 C16:
 MOV AH,9
LEA DX,p
 INT 21H
  JMP OP1
```

```
401     LEA DX,o
402      INT 21H
403       JMP OP1
404
405      C16:
406      MOV AH,9
407     LEA DX,p
408      INT 21H
409       JMP OP1
410
411      C17:
412      MOV AH,9
413     LEA DX,q
414      INT 21H
415       JMP OP1
416
417      C18:
418      MOV AH,9
419     LEA DX,r
420      INT 21H
421       JMP OP1
422
423      C19:
424      MOV AH,9
425     LEA DX,s
426      INT 21H
427
428      JMP OP1
429
430      C20:
431      MOV AH,9
432     LEA DX,t
433      INT 21H
434       JMP OP1
435
436      C21:
437      MOV AH,9
438     LEA DX,u
```

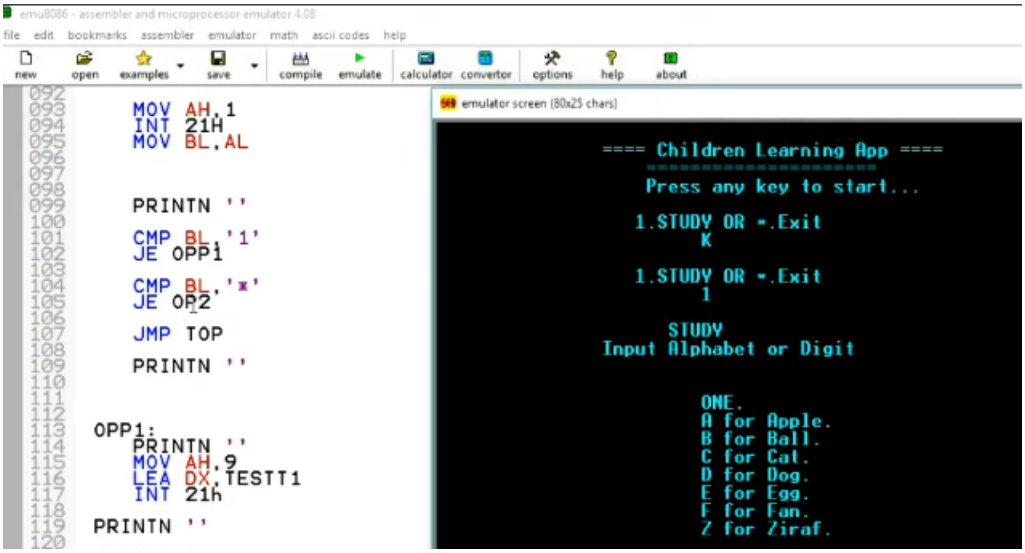
```
435
436     C21:
437     MOV AH,9
438    LEA DX,u
439     INT 21H
440      JMP OP1
441
442     C22:
443     MOV AH,9
444    LEA DX,v
445     INT 21H
446      JMP OP1
447
448     C23:
449     MOV AH,9
450    LEA DX,w
451     INT 21H
452       JMP OP1
453
454     C24:
455     MOV AH,9
456    LEA DX,x
457     INT 21H
458      JMP OP1
459
460     C25:
461     MOV AH,9
462    LEA DX,y
463     INT 21H
464       JMP OP1
465
466     C26:
467     MOV AH,9
468    LEA DX,z
469     INT 21H
470     JMP OP1
471
472         C27:
473    MOV AH,9
```

```
 MOV AH,9
LEA DX,z7
 INT 21H
 JMP OP1

    C34:
 MOV AH,9
LEA DX,z8
 INT 21H
 JMP OP1

    C35:
 MOV AH,9
LEA DX,z9
 INT 21H
 JMP OP1

    C36:
 MOV AH,9
LEA DX,z10
 INT 21H
 JMP OP1

OP2:

PRINTN ''
PRINTN ''

 MOV AH,9
 LEA DX,TNX
 INT 21H

 MOV AH,4CH
 INT 21H

 MAIN ENDP
END MAIN
```

Chapter 3

Performance Evaluation

3.1 Results Analysis/Testing



3.2 Results Overall Discussion

Assembly language is a good language for children to learn. First, assembly language is very concrete. Each instruction in assembly language corresponds to a specific operation that the computer can perform. This can help children to understand how computers work at a fundamental level. Second, assembly language is very challenging. Learning assembly language can be a difficult task, but it can also be very rewarding. When children are able to successfully write a program in assembly language, they can feel a great sense of accomplishment.

Chapter 4

Conclusion

4.1 Discussion

Teaching children the alphabet and digits using assembly language presents both challenges and opportunities:

1.Understanding Fundamentals: Introducing assembly language to kids can offer a deep understanding of how computers work at a fundamental level.

2.Visualizing Concepts: Teaching the alphabet and digits in assembly language can be accompanied by visuals or animations showing the conversion of characters to their binary representations. This visual aid can aid comprehension.

3.Building Problem-Solving Skills: Working in assembly language can enhance problem-solving skills. Teaching children how to write simple programs to display .

4.Building Problem-Solving Skills: Working in assembly language can enhance problem-solving skills. Teaching children how to write simple programs to display

4.2 Scope of Future Work

In future we can more update in this project. We can change or add mane system which is gave us more benefits.