

# Dvx

Tristan B. Kildaire

September 5, 2019

---

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	What is Dvx and why? . . . . .	2
<b>2</b>	<b>Specification</b>	<b>3</b>
2.1	Machine specifications . . . . .	3
2.2	Registers . . . . .	4
2.2.1	General purpose registers . . . . .	4
2.2.2	Special purpose registers . . . . .	5
2.2.3	The flags register . . . . .	5
2.3	Instruction set . . . . .	6

# 1 Introduction

## 1.1 What is Dvx and why?

**Dvx** is a project to experiment with the ideas I have of an architecture that I would like to implement oneday. This is only the beginning of many experiments to be done but basically I wanted to create something in my spare time that I could show my friends. Something that showed them a step into what I would want to do in the future and how I envisage it but atleast in a more tangible way than just some concept (which I already find fine as I know I would be able to implement it anyways).

## 2 Specification

### 2.1 Machine specifications

The machine specifications are layed out below:

1. 8-bit, 16-bit and 32-bit registers available.
2. 32-bit addressing.
3. 32-bit instructions
4. Real mode (MMU nops) and a paged, protected mode (MMU active)
5. Little-endian (reason being that I am writing this specifically to run on x86\_64).

## 2.2 Registers

This section lists all the registers available on the machine.

### 2.2.1 General purpose registers

All of the below registers are general purpose meaning they have no particular meaning to the machine.

There are some exceptions though such as the instructions for stack manipulation.

Name	ID	Size (bytes)	Description
a	0	1	
b	1	1	
c	2	1	
d	3	1	
e	4	1	
f	5	1	
g	6	1	
h	7	1	
ae	8	2	
be	9	2	
ce	10	2	
de	11	2	
ee	12	2	
fe	13	2	
ge	14	2	
he	15	2	
aex	16	4	
bex	17	4	
cex	18	4	
dex	19	4	
eex	20	4	
fex	21	4	
gex	22	4	
hex	23	4	

### 2.2.2 Special purpose registers

These registers and their values have specific meaning to the machine's state.

Name	ID	Size (bytes)	Description
<b>flags</b>	24	4	See flags
<b>ip</b>	25	4	Address of the instruction being executed
<b>vector_table</b>	26	4	Address of the base of the exception table

### 2.2.3 The flags register

The flags register is describe din detail below.


## 2.3 Instruction set

This section describes all of the instructions available in the Dvx architecture.

Name	Op-code	Arguments	Description	Side-effects
<b>nop</b>	0		Does nothing (no-operation)	
	1		Letterlik husidans	
<b>movb</b>	2		Moves immediate byte into byte-wide register.	
<b>movw</b>	3		Moves immediate short/word into short/word-wide register.	
<b>movu</b>	4		Moves the immediate short/word into the upper short/word of the double word-wide register.	
	5			
	6			
	7			
	8			
<b>int</b>	9	<b>register ID</b>	Raises an interrupt with ID held in <b>register ID</b>	Register <b>hex</b> set to the address of the instruction following the int instruction's address.
<b>iret</b>	10		Jumps back to the instruction at the value of <b>hex</b> .	If in protected mode then ring will be set back to 1.
<b>halt</b>	11			