JanskyProcessor Architecture

Ian Duncan

March 21, 2015

Contents

Introduction

This document aims to (eventually) be a complete reference to the JanskyProcessor architecture, instruction set, and computer. I know the name is horrible, but otherwise the project would be nameless. Also, please pronounce the name "yahn-skee" (janski in IPA).

1 Data Storage

The JanskyProcessor computer has a Von Neumann architecture. Program and data memory are both stored in RAM. There also exists thirteen registers, some of which are used for special purposes such as the stack and instruction pointers.

All data is stored in the little-endian format [1]

1.1 Data Size

The JanskyProcessor architecture is a 32-bit architecture. Although it is not required, in practice data is 32-bit aligned in memory, as the `put` and `cpy` instructions operate on 32 bit sections of memory.

Below is a table with the words used to refer to different sizes of data.

Name	**Size in Bytes**
Byte	8
WORD	16
DWORD	32

[1] The emulator does not currently work on big-endian machines due to this fact.

1.2 Registers

Below is a table of all the registers and their properties and uses.

Name	Hex ID	Can Read?	Can Write?	Use
`ar1`	`0x00`	Yes	Yes	Integer arithmetic
`ar2`	`0x01`	Yes	Yes	Integer arithmetic
`ar3`	`0x02`	Yes	Yes	Integer arithmetic
`ar4`	`0x03`	Yes	Yes	Integer arithmetic
`ar5`	`0x04`	Yes	Yes	Integer arithmetic
`ip`	`0x05`	Yes	Only with `jmp` instructions	Pointer to next instruction
`bp`	`0x06`	Yes	Yes	Pointer to instruction to return to (not used currently)

1.3 RAM

RAM is used to store both program and data memory. The JanskyProcessor architecture uses a 32-bit flat addressing model, limiting the memory size to 4 GB.

The first program to be executed is always loaded at address `0x00000000`.

1.4 The Stack

The stack always starts at the highest addressable location in memory (i.e. if RAM is 4096 bytes long, the stack starts at address `0xFFF`), and continues until the stack base. Each item on the stack is always a DWORD.