Journey to the Center of the HP28C/S

by Travis Goodspeed
<travis at utk.edu>

regarding the work of
Paul Courbis
Sébastien Lalande
Disclaimer

I am just the translator. My French sucks.
HP 28S

Photo by Kurt Moerman
Part 1: Software

- Objects: List and Structure
- I/O Zone
- RAM
- Accessing Machine Language
- Further Discoveries
- Useful Routines
Introduction

- Three Models
  - HP28C-1BB
  - HP28C-1CC
  - HP28S-2BB
- HP SATURN architecture, same as HP 71b
- Nibble-wise Little Endian
## Memory Layout

<table>
<thead>
<tr>
<th></th>
<th>HP 28C</th>
<th></th>
<th>HP 28S</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6 Ko</td>
<td>34 Ko</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>60000</td>
<td>52000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RAM</td>
<td>4F000</td>
<td>4F000</td>
<td>RAM</td>
<td>60000</td>
</tr>
<tr>
<td>Void</td>
<td>40800</td>
<td>40800</td>
<td>Void</td>
<td>40000</td>
</tr>
<tr>
<td>Screen</td>
<td>40000</td>
<td>40000</td>
<td>Screen</td>
<td>40000</td>
</tr>
<tr>
<td>Timer</td>
<td>00000</td>
<td>00000</td>
<td>Timer</td>
<td>00000</td>
</tr>
<tr>
<td>ROM</td>
<td>00000</td>
<td>00000</td>
<td>ROM</td>
<td>00000</td>
</tr>
</tbody>
</table>

### Standard

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ROM</td>
<td>00000</td>
<td>00000</td>
<td>ROM</td>
<td>00000</td>
</tr>
<tr>
<td>RAM</td>
<td>4F000</td>
<td>4F000</td>
<td>RAM</td>
<td>60000</td>
</tr>
<tr>
<td>Void</td>
<td>40800</td>
<td>40800</td>
<td>Void</td>
<td>40000</td>
</tr>
<tr>
<td>Screen</td>
<td>40000</td>
<td>40000</td>
<td>Screen</td>
<td>40000</td>
</tr>
<tr>
<td>Timer</td>
<td>00000</td>
<td>00000</td>
<td>Timer</td>
<td>00000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Registers

<table>
<thead>
<tr>
<th>I/O</th>
<th>OUTPUT</th>
<th>INPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Status</th>
<th>CARRY</th>
<th>HW Status</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>4</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pointers</th>
<th>P</th>
<th>D0,D1</th>
<th>PC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20</td>
<td>20 each</td>
<td>20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Safeguard</th>
<th>RSTK</th>
<th>R0 to R4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20 by 8</td>
<td>64 each</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Working</th>
<th>A to D</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64 each</td>
</tr>
</tbody>
</table>
Nybbles (Quartets)

- **Byte**
  - 0x00 to 0xFF
  - 0 to 255

- **Nibble**
  - 0 to F
  - BCD
    - 0 to 9
Nybble Little Endian

Big Endian

Little Endian

HP 28

#1234

12 34

34 12

4 3 2 1
Objects: List and Structure

- **Prologue**
  - 5 nibbles
  - Specifies Type

- **Type**
  - System or User
  - Implies Length

<table>
<thead>
<tr>
<th>Object</th>
<th>Prologue</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Integer¹</td>
<td>#02991</td>
<td>10</td>
</tr>
<tr>
<td>Real</td>
<td>#02933</td>
<td>11</td>
</tr>
<tr>
<td>Extended Real¹</td>
<td>#02955</td>
<td>11</td>
</tr>
<tr>
<td>Complex</td>
<td>#02977</td>
<td>11</td>
</tr>
<tr>
<td>Extended Complex¹</td>
<td>#0299D</td>
<td>12</td>
</tr>
<tr>
<td>Byte¹</td>
<td>#029BF</td>
<td>12</td>
</tr>
<tr>
<td>First Unknown Type¹</td>
<td>#029E1</td>
<td>12</td>
</tr>
<tr>
<td>Array</td>
<td>#02A0A</td>
<td>13</td>
</tr>
<tr>
<td>Second Unknown Type¹</td>
<td>#02A2C</td>
<td>13</td>
</tr>
<tr>
<td>String</td>
<td>#02A4E</td>
<td>14</td>
</tr>
<tr>
<td>Binary Integer</td>
<td>#02A70</td>
<td>14</td>
</tr>
<tr>
<td>List</td>
<td>#02A96</td>
<td>15</td>
</tr>
<tr>
<td>ROM/RAM Pair¹</td>
<td>#02AB8</td>
<td>15</td>
</tr>
<tr>
<td>Algebraic</td>
<td>#02ADA</td>
<td>19</td>
</tr>
<tr>
<td>Program</td>
<td>#02C67</td>
<td>21</td>
</tr>
<tr>
<td>Assembly Code¹</td>
<td>#02C96</td>
<td>22</td>
</tr>
<tr>
<td>Global Name</td>
<td>#02D12</td>
<td>22</td>
</tr>
<tr>
<td>Local Name¹</td>
<td>#02D37</td>
<td>22</td>
</tr>
<tr>
<td>ROM Pointer¹</td>
<td>#02D5C</td>
<td>23</td>
</tr>
</tbody>
</table>

Table 2.1: Object Types and their Prologues
Short Integer
#02911

<table>
<thead>
<tr>
<th>Prologue</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>11920</td>
</tr>
<tr>
<td>12345</td>
<td>00000</td>
</tr>
<tr>
<td>11920</td>
<td>54321</td>
</tr>
</tbody>
</table>
### Real

**#02933**

<table>
<thead>
<tr>
<th>Prologue</th>
<th>Exponent</th>
<th>Mantissa</th>
<th>Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>#02933</td>
<td>5</td>
<td>3.14159...</td>
<td>+</td>
</tr>
</tbody>
</table>

\[ \pi \times 10^5 \]

| 33290 | 500 | 953562951413 | 0    |
Reverse Polish Notation

- Stack Metaphor
- Left to right.
  - If value
    - PUSH onto stack.
  - If operator,
    - POP parameters.
    - PUSH result.
- No grouping symbols.
- Easy to interpret.

\[ 1 + 10 \times 3 \]
RPN Interpretation

Input

Stack

10 3 * 1 +

10 3 * 1 +

|
1

31
Algebraic #02ADA

Prologue

Expression

End. Alg.

$\pi \times 10^5$

$\pi \times 10^5$

$\pi \times 10^5$

$\pi \times 10^5$
Algebraic
#02ADA

\[ \pi \times 10^5 \]
Algebraic
#02ADA

\[ \pi \times 10^5 \]
π \times 10^5
I/O Zone

- IR in/out
- Timers
- Contrast
- Screen/Indicators
- Speed
  - 28S Only
- Row Driver Waveform
  - 28C Only
Row Driver Waveform

- Maps logical lines to physical lines.
- Scrolling
  - Swap line-mappings, not data.
- Mirroring
  - Map the same logic line to many physical lines.
- Initialized at boot.
- Interpreted by software video driver.
Row Driver Waveform
RAM

- Polling
  - Background Processes Watch RAM
  - I/O-like Behavior
- Keyboard Buffer
- Flags
- Command-Line
- Stacks
- Temp Environment
Keyboard Buffer

- Samples Keys
  - Not ASCII
- Clock
- KEYPSTART
  - next to be handled
- KEYEND
  - next to be sampled
Accessing Machine Language

- **ML Program**
  - is an Object Type (#02C96)
  - may not be created by the user.

- **Bootstrapping**
  - ASS
  - LASS
LASS

- Takes as input an ML string.
- One character, one nybble.
- Assembles
  - ML String to ML Object
  - not Assembly String to ML Object
- Usage
  - Push string to stack.
  - LASS
  - Store resultant object to variable.
INV.VID
<table>
<thead>
<tr>
<th>INV.VID</th>
</tr>
</thead>
<tbody>
<tr>
<td>76C20</td>
</tr>
<tr>
<td>91C70</td>
</tr>
<tr>
<td>69C20</td>
</tr>
<tr>
<td>E5000</td>
</tr>
<tr>
<td>13310</td>
</tr>
<tr>
<td>334FD</td>
</tr>
<tr>
<td>2041F</td>
</tr>
<tr>
<td>87004</td>
</tr>
<tr>
<td>72203</td>
</tr>
<tr>
<td>4FD50</td>
</tr>
<tr>
<td>41F00</td>
</tr>
<tr>
<td>40470</td>
</tr>
<tr>
<td>10113</td>
</tr>
<tr>
<td>13114</td>
</tr>
<tr>
<td>21648</td>
</tr>
<tr>
<td>08C27</td>
</tr>
<tr>
<td>1531B</td>
</tr>
<tr>
<td>9C151</td>
</tr>
<tr>
<td>11771</td>
</tr>
<tr>
<td>33131</td>
</tr>
<tr>
<td>8BA9E</td>
</tr>
</tbody>
</table>

- “76C2091C7..."
- LASS
- `INV.VID' STO
Further Discoveries

- **Search ROM**
  - Strings Tables in Appendix
  - I/O or RAM addresses
- **Systematic Disassembly**
  - Disassembler Program
  - Dump to a Workstation
Useful Routines

- SAV.REG
- LOAD.REG
- RES.ROOM
- GARB.COLLM
  - Force Garbage Collection
- TFM
  - Too Few Memory
- ERROR
- BEEP
Useful Routines

SAV.REGN

C=reserved adr

RES.ROOM

carry=0

carry=1

GARB.COLL

First Try?

yes

no

LOAD.REG

continue

LOAD.REG

TFM
Part 2: Hardware

- Exterior Description
- Opening the Machine
- Interior Description
- Transformations
  - External Power Feed
  - Overclocking
  - Extended Memory
- The I/O Port
- Closing the Machine
- I/O Interfaces
Exterior Description

- Keyboard
- Electronics
- Keyboard
Opening

- No Screws!
- Gotta break some pins.
Interior Description
Transformations

- External Power Feed
- Overclocking
- Memory Upgrade
Overclocking

- **HP28C**
  - Replace or Remove a Capacitor
  - Replace an Inductor

- **HP28S**
  - Overclocking in Software
  - SPEED Program
Memory Upgrade

- HP71B Memory Module
- Installation
  - Cut Traces of Memory Bus
  - Patch in Upgrade Module
I/O Ports

- Hack of the IR Led

- Hack of the Buzzer
I/O Ports

- IR Cartridge
- Replacement Adapters
  - Apple ][e Joystick
  - HP to HP
  - RS232
HP28 Printer

- HP82240B
Ideas!

- Add a Joystick
- Telephone Composer
  - Photocoupler and Relays
  - Provides click of a rotary phone.
  - Dial by counting digits, then waiting.
- Motor Control
- Robot
- Plotter
Part 3: Appendices

- Machine Language
- SATURN Microprocessor
- SATURN Instructions
- Exhaustive List of Error Codes
- List of Objects and RPL Entry Points
- Library of Programs
HP28 Emulation

- Christoph Giesselink
HP48 Series

- Memory Expansion
- IR I/O
- Serial Port
- Meta Kernel
  - Replacement OS

Photo by Heretiq
HP49 Series

- SD Card Slot
- USB, IR, Serial
- ARM CPU
  - Emulates SATURN
- HPGCC
- Meta Kernel
  - Official OS

Photo by Nova dc
Next Step: TI Nspire

- No ML Programming
- No Games
- No ROM Dump
Questions?

Travis Goodspeed
travis@utk.edu
865.300.0094