Using the IIgs Monitor

Stavros Karatsoridis
KFest 2008
Monitor?

Monitor?
No. Monitor!
Objectives

- Learn how to use the Monitor program in the IIgs to
  - Type in programs
  - Examine and change memory
  - Execute programs
  - Perform miscellaneous tasks
Entering the Monitor

- From the AppleSoft BASIC prompt, type:
  ```call -151```
- Using the Visit Monitor CDA
- (More on that later)
Displaying Memory

- `{Addr}` displays the value at memory location `{Addr}`.
- `{Addr1}.{Addr2}` displays a block of memory starting at `{Addr1}` and ending at `{Addr2}`.
Modifying Memory

‣ \{Addr\}:\{Val\}
  Stores the value \{Val\} at address \{Addr\}

‣ \{Addr\}:\{Val1\} \{Val2\} ... \{ValN\}
  Stores values in consecutive areas of memory starting at \{Addr\}

‣ This is one of the main ways you enter programs into memory

‣ \{Addr\}:\{“ASCII string”\}
  Places the ASCII string in memory starting at \{Addr\}
Moving Data

M (Move)

\{Dest\}<\{From1\}.\{From2\}M

Moves (copies) the range of memory starting at \{From1\} and ending at \{From2\} to \{Dest\}

Z (Zap)

\{Val\}<\{From1\}.\{From2\}Z

Fills the range of memory starting at \{From1\} and ending at \{From2\} with \{Val\}
Several ways to search a range of memory

**Values**

\{Val1\} ... \{ValN\}\<\{From1\}.\{From2\}P

Searches the range of memory from \{From1\} to \{From2\} for the values \{Val1\} to \{ValN\} in that order.

**Strings**

\{“ASCII String”\}\<\{From1\}.\{From2\}P

Searches the range of memory from \{From1\} to \{From2\} for the values in that order.
Disassembler

- `{Addr}L`
  - Disassembles the next 20 commands (opcodes) starting at `{Addr}`
  - Subsequent entries of “L” by itself will display the next 20 commands and so on
  - May use several Ls on a line to display more than 20 commands at a time (E.G. `{Addr}LLL`)
Running Programs

- G (Go)
- {Addr}G

Runs the program starting at {Addr}
Use the monitor to enter the following program into memory, starting at $300

- 300- A2 05
- 302- 20 DD FB
- 305- CA
- 306- F0 03
- 308- 4C 02 03
- 30B: 60
Registers


‣ \{\text{Control-E}\}
  ‣ Displays the contents of the 65816’s registers

‣ Each register can be changed directly
  ‣ \{\text{Val}\}=A \text{ (Changes Accumulator to } \{\text{Val}\})
  ‣ \{\text{Val}\}=X \text{ (Changes X register to } \{\text{Val}\})
  ‣ \{\text{Val}\}=Y \text{ (Changes Y Register to } \{\text{Val}\})
Hexadecimal/Decimal Conversion

To convert a value from decimal to hexadecimal

\[ \{\text{Dec}\} = \]

To convert a value from hexadecimal to decimal

\[ = \{\text{Hex}\} \]
You can perform hexadecimal calculations with the following:

- `{Val1} + {Val2}` (Addition)
- `{Val} - {Val2}` (Subtraction)
- `{Val1} * {Val2}` (Multiplication)
- `{Val1} _ {Val2}` (Division)
Hands On

Perform the following calculations. What do you get for each one?

- $6 \times 4$
- $\text{BEEF} \div 5$
- $\text{BAD} + \text{DAD}$
- $\text{DEAF} - \text{DEAD}$
Other Commands

› I (Inverse Video)
  › Sets display mode to inverse video (Like "INVERSE" command in BASIC)

› N (Normal Video)
  › Sets display mode to normal video (Like "NORMAL" command in BASIC)

› {Slotnum}{CTRL-P}
  › Redirects output to {Slotnum} (Like "PR#" in BASIC)

› {Slotnum}{CTRL-K}
  › Redirects input from {Slotnum} (Like "IN#" in BASIC)
Other Commands

- =T
  - Displays the Time and Date

- =T=mm/dd/yy hh:mm:ss
  - Changes time and date to that specified by mm/dd/yy hh:mm:ss
Other Commands

‣ # (Install CDAs)
  ‣ Installs two useful CDAs for debugging purposes
  ‣ Visit Monitor
  ‣ Memory Peeker
Other Commands

‣ Q (QUIT)
  ‣ Quits monitor program
  ‣ Same as typing 3D0G

‣ {CTRL-C}
  ‣ Jumps to BASIC (Warmstart)
Toolbox Calls

- You can call the toolbox from the monitor
- U (Tool Locator)
- Described in detail in Apple IIgs Firmware Reference manual
Example

\C 2 0 0 0 1 20 81 0FF 0 8D 0 1 24 C\U
<return>

Enter some text afterwards. The monitor will respond by printing the hex count of the number of characters.

Store the value given at location 01/2080

\4 0 0 1 20 80 1C C\U <return>

Displays the text you entered in step 1
The Mini-Assembler is a way to enter programs using mnemonic names for the opcodes instead of using the numeric values.

The command to start the Mini-Assembler is ! (Exclamation Point)
Use the Mini-Assembler to enter the following program

```
00/300:LDX #00
 LDA $30E,X
 BEQ $30D
 JSR $FDED
 INX
 BRA $302
 RTS
"KFEST 2008
:00
```
Questions?