

ARM7 Firmware Engineering

**Silicon Valley Forth Interest Group
Presented**

**By
Dr. C. H. Ting
eForth Technology, Inc.**

September 25, 2004



Summary

- **ARM7 on GameBoyAdvance**
- **ARM7 on ADuC7024**
- **Arm7 eForth v1.01 based on Win32Forth for GBA**
- **ARM7 eForth v5.02**
- **Porting eForth 5.02 to GBA**
- **ARM7 32-bit Instruction Set**
- **ARM7 Assembler**
- **ARM7 Subroutine Thread Model**
- **ARM7 Metacompiler based on F#**

ARM7 on GameBoyAdvance

- **16 MHz ARM7 core**
- **32 Kbytes internal RAM**
- **256 Kbytes external RAM**
- **32 Mbytes Flash RAM**
- **240x160 Color Display**
- **10 Switches for user interface**
- **Serial Communication Port**
- **Graphic objects**
- **Sound objects**

ARM7 on ADuC7024

- **45 MHz ARM7 core**
- **64 Kbytes Flash RAM**
- **8 Channels of 12-bit A/D**
- **4 Channel 12-bit D/A**
- **Serial Port**
- **Parallel Port**
- **Counters, Timers, Interrupt Controller**
- **Keil Program Development System**

Arm7 eForth v1.01 based on Win32Forth for GBA

- **Original Direct Thread eForth Model**
- **Assembler derived from FPC assembler**
- **Ported to several ARM7 platforms**
- **Extensive Applications for GBA**
 - **Chinese Character Generator**
 - **eBooks**
 - **Bilingual Bible**
 - **DSO Simulator**

ARM7 eForth v5.02

- **Implemented on Analog Devices ADuC7024**
- **Based on eForth v2.0 using Subroutine Thread Model**
- **Fully optimized for ARM7 core**
- **Implemented in AS assembler**
- **Intended for Digital Storage Oscilloscope project**

Porting eForth 5.02 to GBA

- **Combine Name and Code Dictionaries**
- **Change from Direct Thread Model to Subroutine Thread Model**
- **Rewrite simplified assembler**
- **Change platform from Win32Forth to F#**
- **Verify all GBA applications**
- **Verify eForth interpreter/compiler**

ARM7 32-bit Instruction Set

- **Conditional execution field**
- **Register fields**
- **Immediate and offset field**
- **Operation field**
- **Miscellaneous bits**

ARM7 Assembler

- **Field Operators deposit bit patterns:**
 - **Conditionals**
 - **Register assignments**
 - **Memory accessing modes**
 - **Immediate values**
 - **Offsets**
- **Postfix Opcode:**
 - **Deposit opcode**
 - **Fix registers if necessary**
 - **Process immediate and offsets if necessary**

ARM7 Subroutine Thread Model

- **ARM7 does not have CALL and RET instructions**
- **It has an one level return stack in the Link Register LP**
- **A subroutine call uses Branch-and-Link instruction**
 - **<offset> ,BL**
- **A subroutine return uses Branch-and-Exchange instruction through LP**
 - **LP ,BX**

ARM7 Subroutine Thread Model

- **All code words are terminated by**
 - LP ,BX
- **All words are referenced by:**
 - <offset> ,BL
- **High level words start with:**
 - [-T] RP, {{ LP }} ,STMFD
- **High level words are terminated by**
 - [-T] RP, {{ PC }} ,LDMFD

ARM7 Metacompiler based on F#

- | | |
|-------------------------|---------------------------|
| ■ GameGirl9.fex | Loader |
| ■ ARM7meta.f | Metacompiler |
| ■ ARM7asm.f | Assembler |
| ■ ARM7kernel.f | eForth Kernel |
| ■ ARM7eforth.f | eForth Interpreter |
| ■ ... | Applications |
| ■ ARM7compiler.f | eForth Compiler |

Thank you very much!

