Booting Without PROMS

Jerry Needell

David Bodet

University of New Hampshire

This presentation contains no United States Export Controlled information.

Basic Concept

- **UNH CDPUs are designed to have a highly reliable Boot segment that allows for system diagnostics and reprogramming without relying on specific memory devices.**
- **Key to recovery from damaged/failed parts.**
- **FSW can be rebuilt to execute from available resources.**

IP Core in FPGA

- **Typical implementation uses PROMS to host Boot program.**
- **For RBSP/MMS the UNH CDPU uses a customized LEON3-FT IP core implemented in an ACTEL RTAX2000 FPGA**
- **AHBROM feature of LEON3 allows implementation of Boot Code within the FPGA**

AHBROM

- **Advanced Highspeed Bus Read Only Memory**
- **Optional configuration for LEON3**
- **MMS/RBSP implementations**
  - ~3 % of Combinatorial Logic-Cells
  - <1 % of Register-Cells

AHBROM

- **Overview (From GRIP Documentation (www.aeroflex.com)**
  - The AHBROM core implements a 32-bit wide on-chip ROM with an AHB slave interface. Read accesses take zero waitstates, or one waitstate if the pipeline option is enabled. The ROM supports byte- and half-word accesses, as well as all types of AHB burst accesses.
- **The AHBPROM is automatically generated by the make utility in GRLIB. The input format is a sparc-elf binary file, produced by the BCC cross-compiler (sparc-elf-gcc). To create a PROM, first compile a suitable binary and the run the make utility:**
  - bash$ sparc-elf-gcc prom.S -o prom.exe
  - bash$ make ahbrom.vhd
    - Creating ahbrom.vhd : file size 272 bytes, address bits 9
- **The created PROM is realized in synthesizable VHDL code, using a CASE statement. For FPGA targets,**
- **most synthesis tools will map the CASE statement on a block RAM/ROM if available.**

KEY FEATURES

- **Requires No External Resources**
  - RAM/NVRAM
- **Allows testing of all interfaces before installing memory devices**
  - Reduces risk of removal
- **Reduces parts count**
  - Eliminates PROMS

FIELDS (MMS) CDPU

CDPU Board Testing

- **CDPU Board Test Sequence**
  - Install all but SRAM/C-RAM
    - Execute Board Functional Test using AHBROM
  - Install SRAM
    - Test SRAM via AHBROM
  - Install C-RAM
    - Load POST to SRAM
    - Test CRAM
    - Load FSW into CRAM (POST and RTEMS application)
  - Execute Full Functional Test

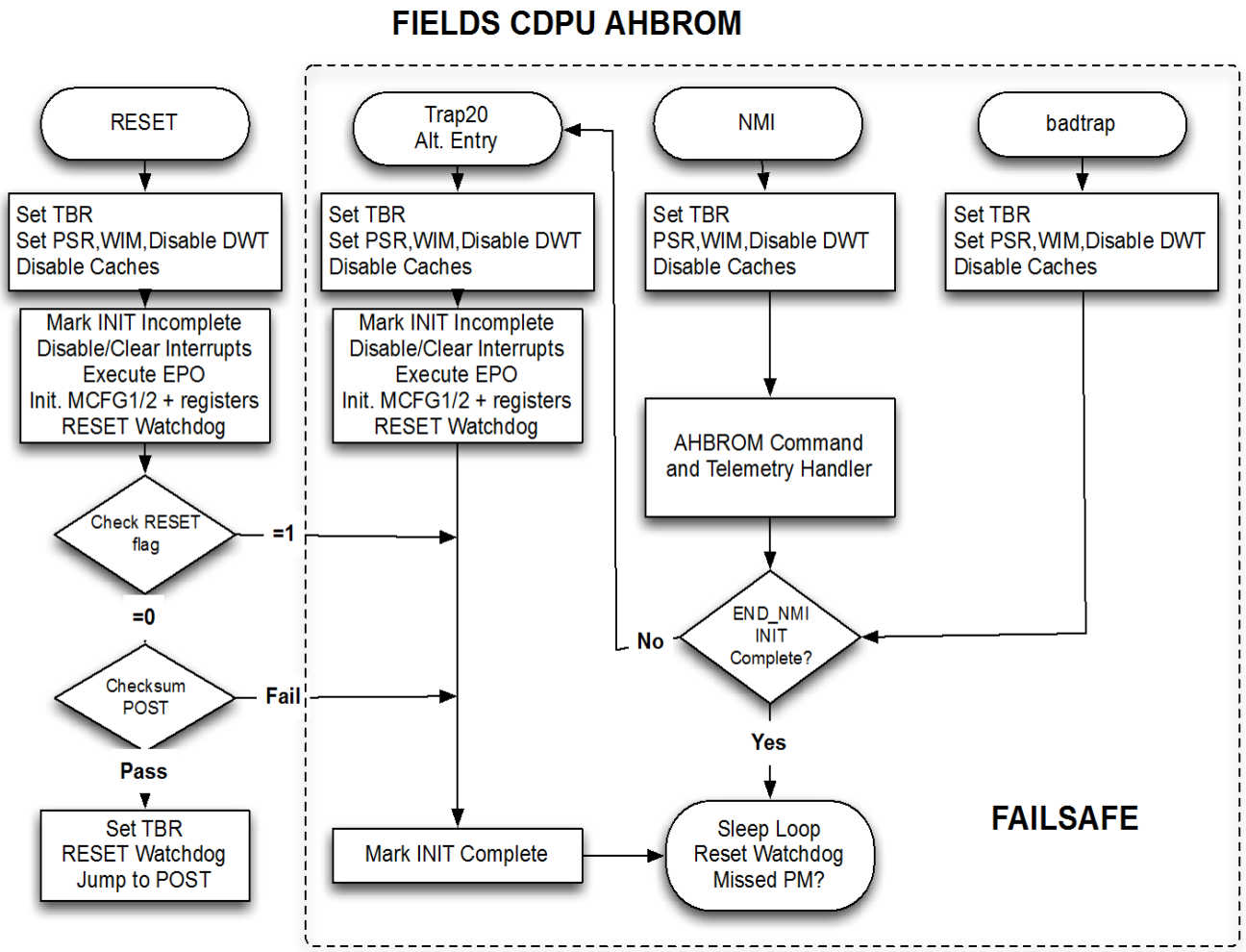
AHBROM BOOT CODE

- **Pass through on RESET**
- **Normal flow**
  - Test CRAM containing POST
  - If OK – Jump to POST
  - If Not, enter FAILSAFE.
- **Alternate path**
  - RESET flag prevents promotion to POST
    - Enter FAILSFE
  - Fault jumps directly to FAILSAFE

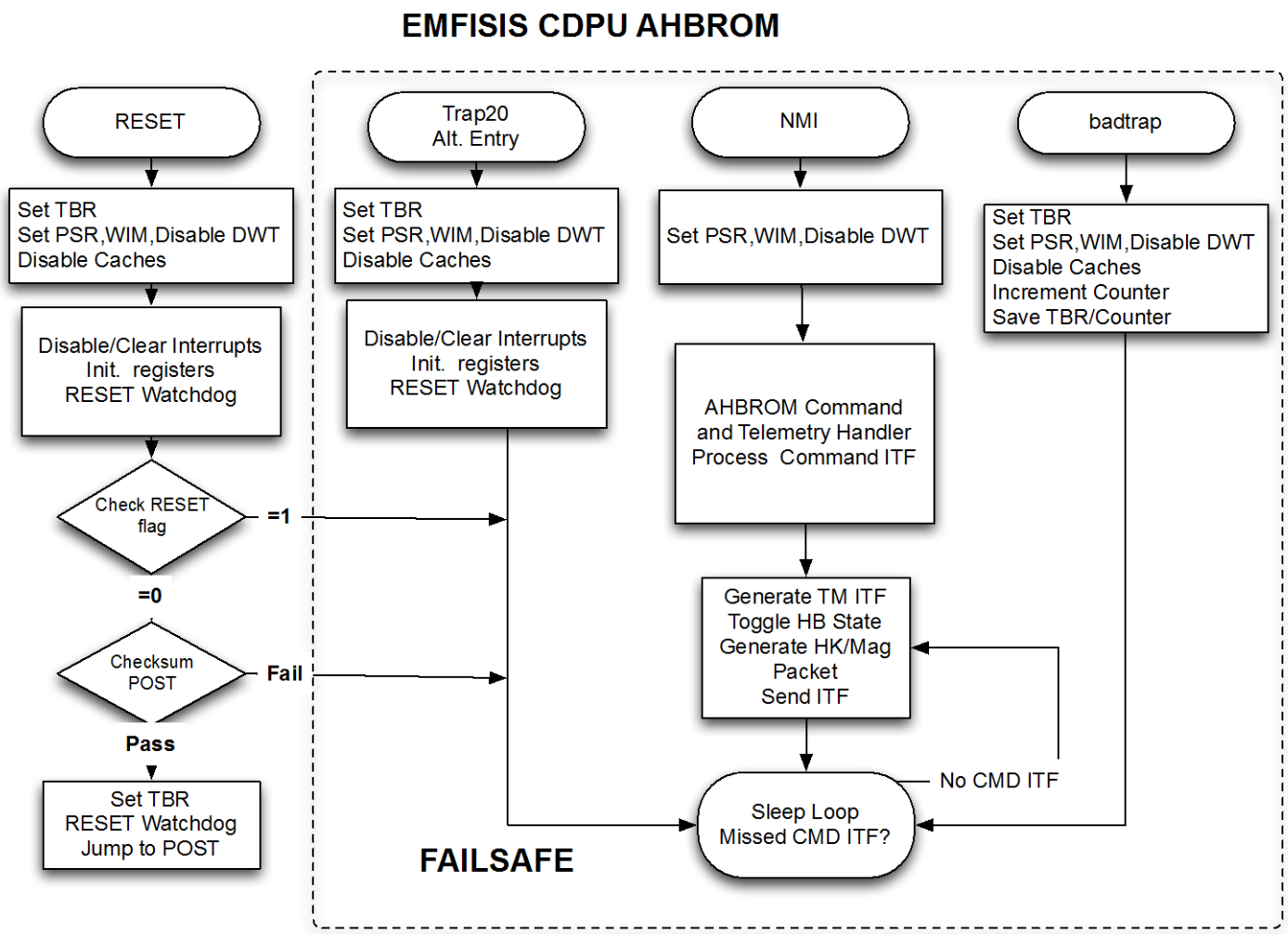
FPGA PARSED COMMANDS

- **RESET**
  - Accepts 8 bit field passed to FSW
  - AHBROM uses on bit to prevent promotion
  - POST uses other bits to select Boot Image
- **C-RAM Write Enable**
  - FSW cannot enable C-RAM write
- **Block “Periodic Message”**
  - Prevents interference with Boot Sequence
  - Enabled by FSW (FAILSAFE or RTEMS)
- **All other commands passed to FSW via NMI**

FIELDS (MMS) AHBROM



EMFISIS (RBSP) AHBROM



FAILSAFE

- **Simple/reliable**
- **Permits flexible response to faults/failures**
- **Can address any Memory device**
- **Supports critical Telemetry**
  - MMS Heartbeat Status
  - RBSP
    - Required ITF
    - Critical Mag data

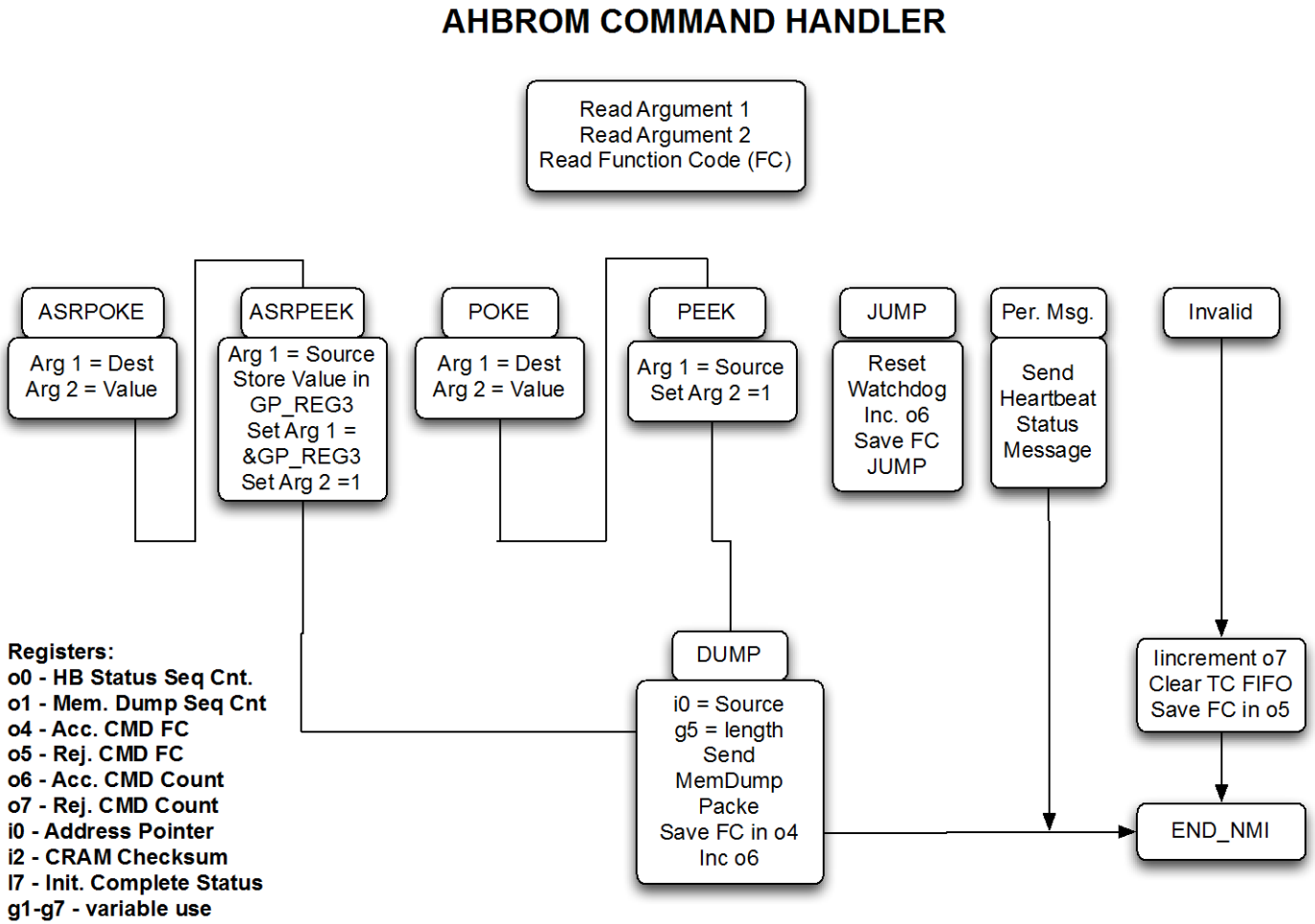
FAILSAFE

- CDPU must be Commandable!
- Commands trigger NMI
- Minimal Command set implemented
- Minimal telemetry
- Meant to be a reliable building block

PEEK/POKE/JUMP

- **PEEK – read any Address**
- **POKE – Write to any Address**
- **JUMP – Execute from any address**
- **Enough to completely reprogram system**
  - Load “load program”
  - Jump to it

FIELDS (MMS) AHBROM COMMAND HANDLER



AHBROM SPECIAL CONSIDERATINS

- **No Stack**
- **Cannot execute RTI**
- **Every Command reinitializes processor**
  - PSR, WIM, TBR
- **Use registers to hold pointers and configuration flags**
- **FPGA provides a few General Purpose registers that are preserved through RESET**

OPERATIONAL EXPERIENCE

- **2 RBSP CDPUs on orbit**
- **3 MMS CDPUs built/ delivered – 1 remaining**
- **Several years of in-situ testing.**
  - Executed every time board turns on!
  - Used to program all EM/Flight FSW
    - No “test” interface
- **Unintended “test” post delivery for MMS FM1**
  - Full “Wipe” and Reload of C-RAM after delivery
  - Execute via flight-like command/telemetry path.