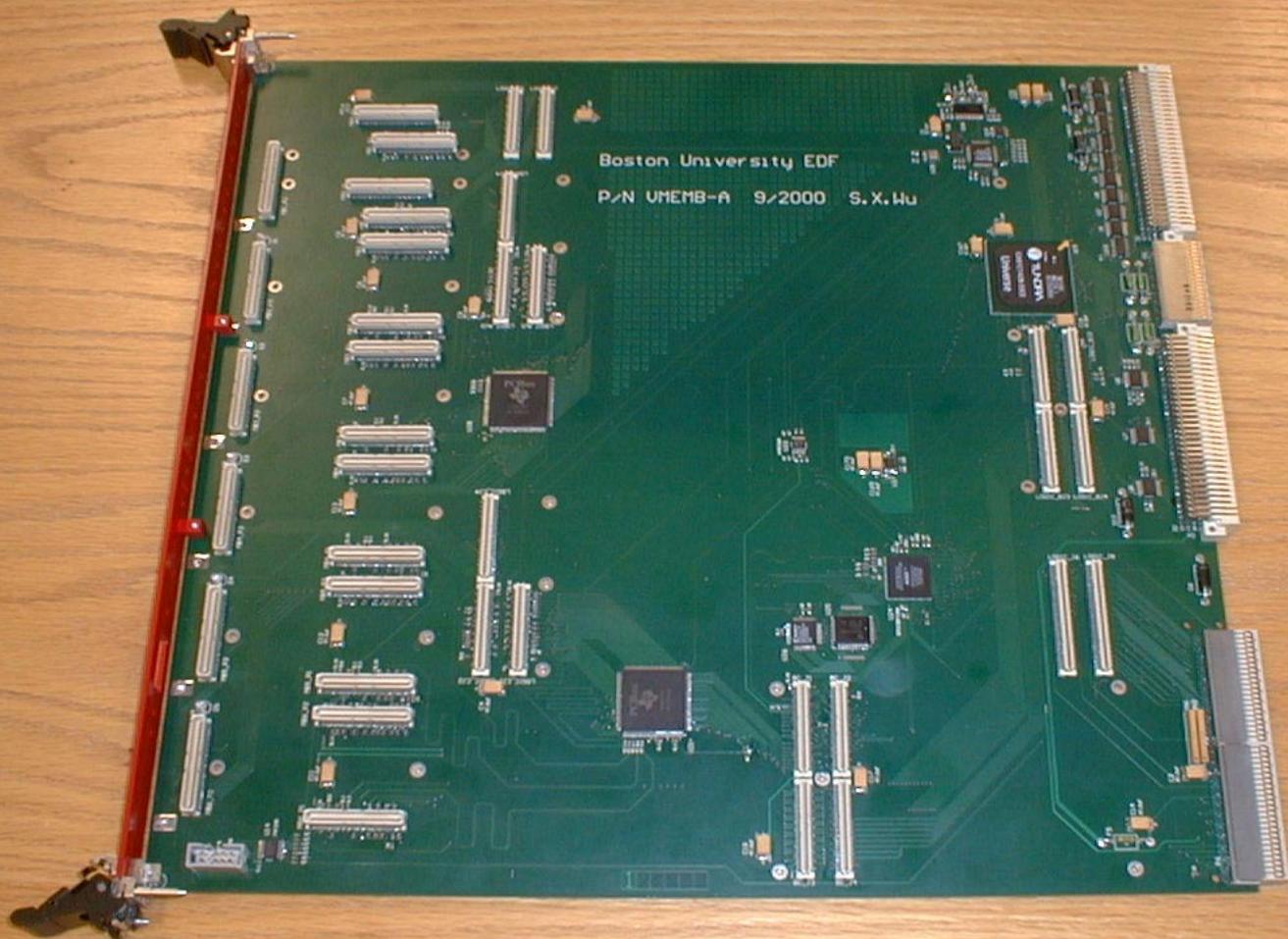


# Status of the STT Motherboard Testing

Evgeny Popkov

Boston University

5 Jan. 2001



Boston University EDF  
P/N UMEMB-A 9/2000 S.X.Hu

# Test Stand

---

## Equipment:

- VME crate with a power supply
- MVME2302 processor with a VT200 console terminal and Ethernet connection
- SGI workstation with VxWorks development suite
- PC running WinNT with Altera MAX+PLUS II

## Procedure:

- The processor boots VxWorks from the SGI disk using ftp
- The SGI disk is mounted from VxWorks
- VxWorks is accessible through the console, telnet and ftp
- A test program is compiled on SGI and loaded to VxWorks

One can compile a program on d0mino and load it to VxWorks through ftp

# Software

---

Axel Naumann wrote a C++ class library which allows to initialize Universe2, initialize PCI devices and do PCI read/write transfers.

A user should provide a driver for his/her PCI device (a driver is written using some standard rules).

In principle the software should detect and initialize all known PCI devices (final goal).

Software is not finished yet and needs more testing.  
Documentation is required.

# Motherboard Testing

---

Tests of the VME - PCI-3 interface:

- The Universe2 is correctly initialized at powerup: the Universe2 Control and Status Registers (CSR) are accessible through VME A24 address space after powerup
- Can initialize Universe2 so that the VME A32 address space is mapped to: the Universe2 CSR registers and configuration registers of the PCI bridge and the local control FPGA on the PCI-3 bus

Tests of the PCI-3 bus:

- Can read/write to the configuration registers of the PCI3 devices
- As of today I cannot access the local control FPGA memory - work with Shouxian to resolve that

# Software Testing

---

Already tested:

- Software compiles and loads ok
- Could not make a standard C function “memcpy” write to the Universe2 registers - program crashes - had to replace several methods with a less generic code (burst transfers are not possible)
- Universe2 initialization code works
- PCI configuration space read/write methods work.

Needs to be done:

- Resolve “memcpy” problem to be able to make burst transfers
- Write a driver for the PCI-to-PCI bridges
- Test/develop the PCI part of the software (basic classes are available)
- Documentation.

# Near term plans

---

- Test single and burst read/write transfers to the FPGA memory on PCI-3
- Program the onboard flash memory for the local control initialization at powerup
- Initialize the PCI-1,2 bridges to access the PCI-1,2 buses
- Check the PCI-1,2 (access to the configuration registers and memory on PMC and PC-MIP cards)
- Determine the clock frequency limit on the PCI-3 bus
- Test/develop software