A Bridge Too Far?

Using a PCI/VME Bridge with EPICS and Porting vxWorks Drivers

Wayne Lewis
Control Systems Group
Australian Synchrotron

Overview

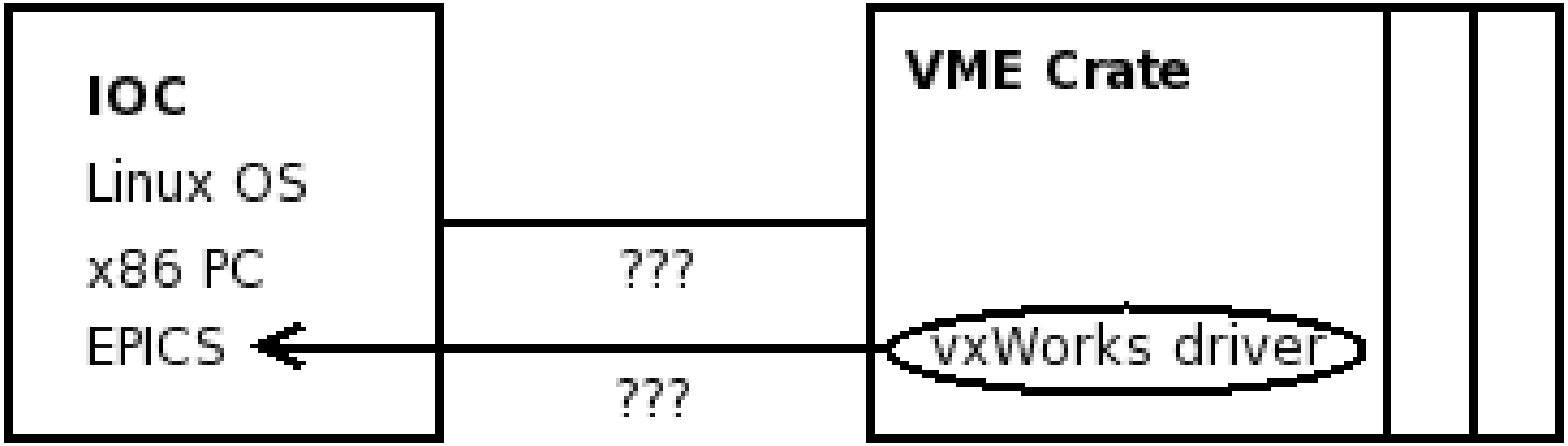
Overview

- Background
- Hardware
- Software design
- Migration of drivers
- Performance
- Benefits
- Further work
- Acknowledgements

The question

Our objectives

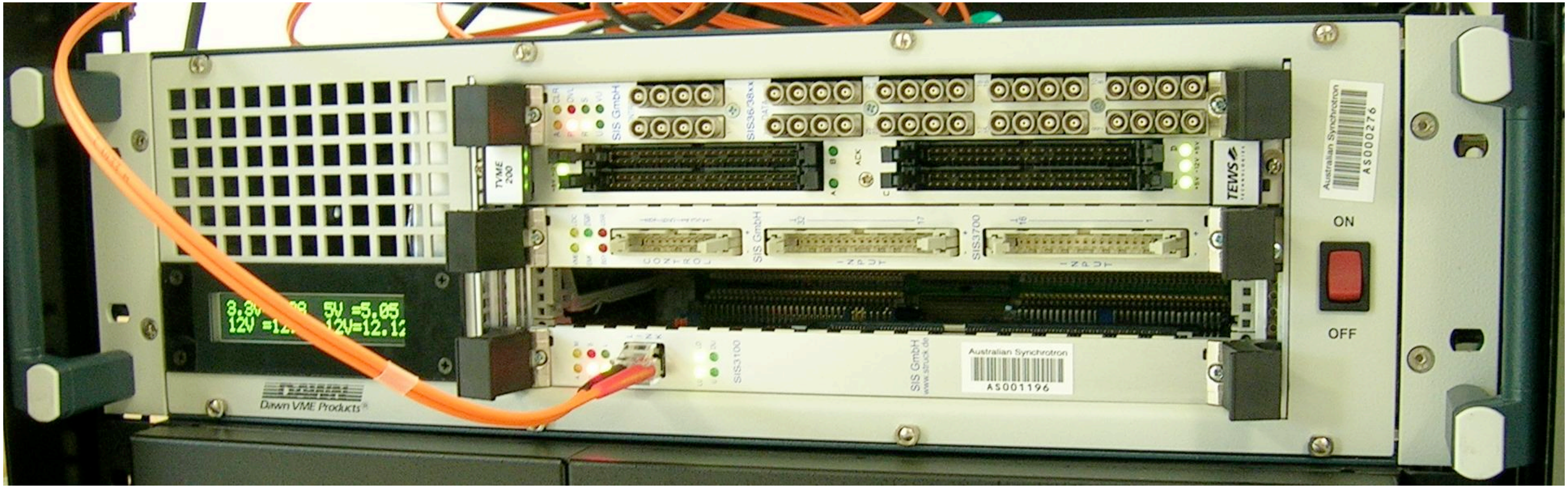
- Use x86 based Linux IOCs
- Use VME data acquisition hardware
- Avoid vxWorks
- Reuse existing drivers for VME hardware with as little modification as possible



A hardware answer

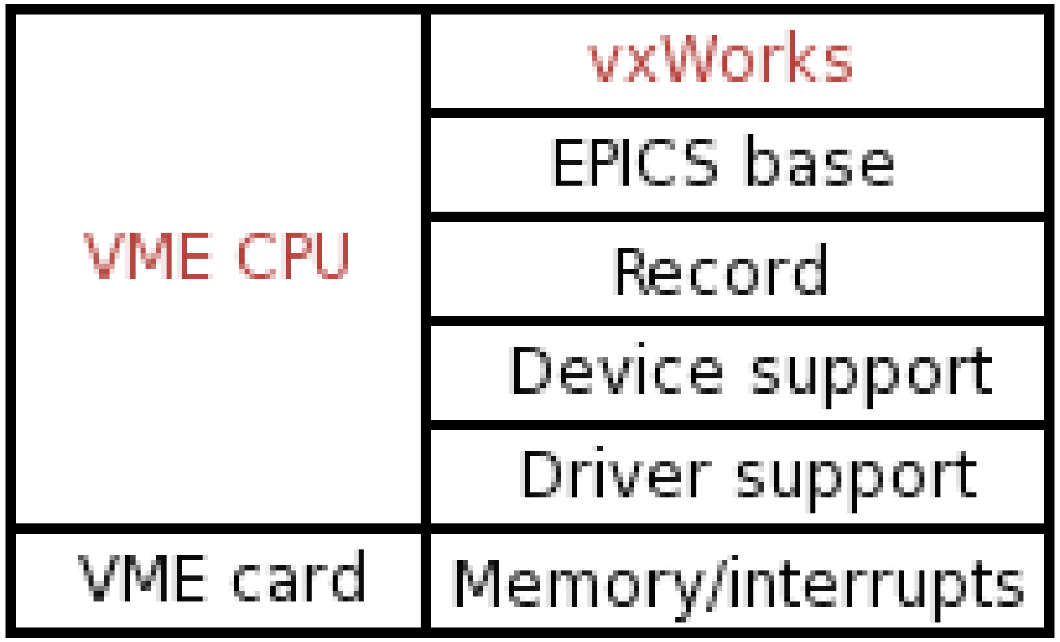
- SIS 1100/3100 by Struck (www.struck.de)
- Already in use at CLS
- Same manufacturer as some chosen data acquisition equipment
- Universal PCI board now available
- VME board replaces VME CPU
- PCI board inserted into any PCI slot in PC
- Connected using fibre optic link

SIS3100 installed in VME crate

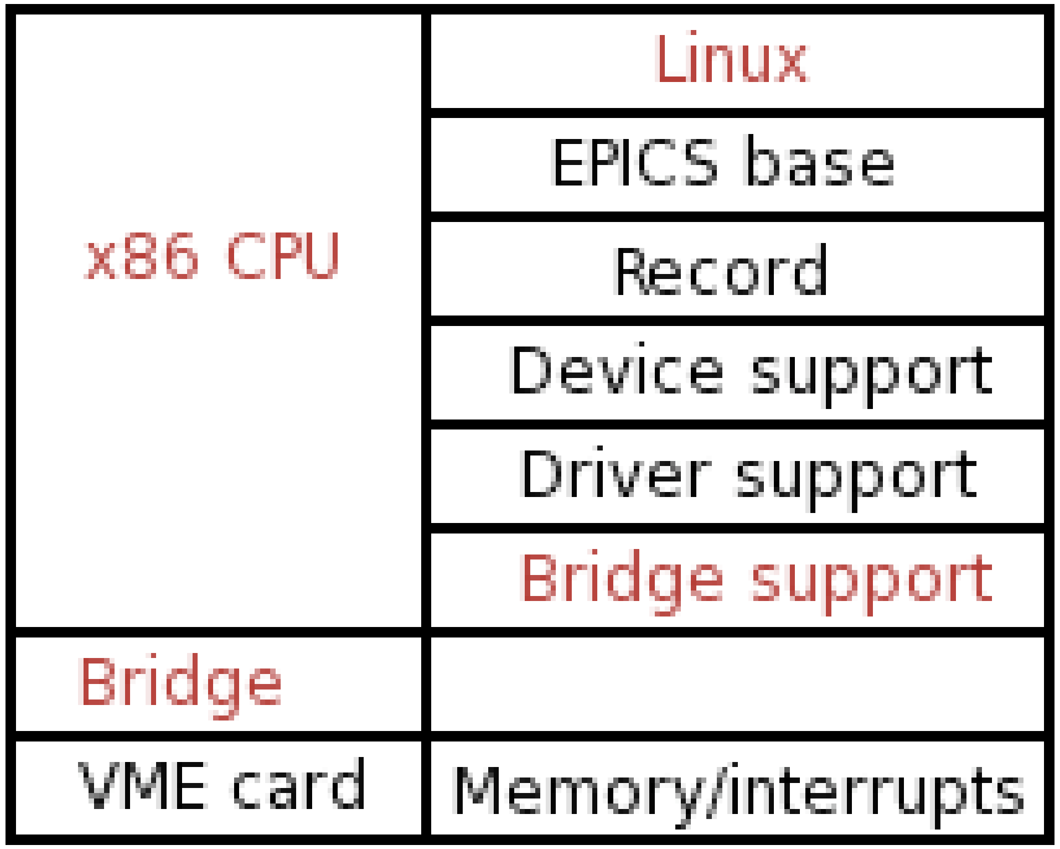


A software answer

- Utilise Operating System Independent elements in EPICS
- Write implementations of devLib routines for Linux using the Struck bridge
    - devLib currently only exists for vxWorks and RTEMS
- Develop code as an EPICS module so that it can be loaded if required
- Modify EPICS base to allow module to supply devLib implementation



VME CPU	vxWorks
	EPICS base
	Record
	Device support
	Driver support
VME card	Memory/interrupts



x86 CPU	Linux
	EPICS base
	Record
	Device support
	Driver support
	Bridge support
Bridge	
VME card	Memory/interrupts

Software design - overall

- Build as an EPICS module to make use transparent
- Memory map to PC memory to mimic operation of VME memory addressing
- VME interrupts converted to Linux signals
- Provide implementations of the EPICS devLib commands to allow Operating System independence:
  - `devMapAddr`
  - `devConnectInterruptVME`
  - other functions are configured to return success at this stage. Not required for this stage of development.

Software design - EPICS integration

- Minor change required to EPICS base to allow a library to provide devLib function implementations.
  - Compatible with existing vxWorks/RTEMS implementations
- Add simple `devLibOSD.c` to $EPICS_BASE/src/libCom/osi/os/default

```
#include <stdlib.h>
#include "devLib.h"
devLibVirtualOS *pdevLibVirtualOS;
```

- Update Makefile for libCom
- Implementation of devLib can be supplied in a module in the EPICS IOC executable.

Software design - memory map

- Memory map to PC memory to mimic operation of VME memory addressing
- Provides implementation of `devMapAddr`
- Can map address either with or without interrupt support - this function needs to be split into the two elements
- Requires a board number - purely used as an identifier.
- Maintains list of boards and VME and PC addresses
- Can access 16, 24 and 32 bit VME address spaces
- Utilises `mmap` library command to map memory

Interrupt handling

Software design - interrupt handling

- VME interrupts are converted to PCI interrupt by SIS hardware and to signal by SIS kernel modulenN
- Use `sigaction` function to register interest in signal and trigger execution of a signal handler
- Maintain a map of VME interrupt vectors and locations of Interrupt Service Routines in PC
- Map is populated as part of card initialisation functions

iocsh commands

- The following iocsh commands are provided to configure the bridge
    - `initPciVmeBridge(crate, controlFile, remoteFile)`
    - `mapMemoryNoIRQ(crate, board, VME address, VME AM)`
    - `mapMemoryWithIRQ(crate, board, VME address, VME AM, interrupt vector, interrupt level)`

Migration of existing vxWorks driver

- Replace header includes
- Replace vxWorks specific commands
- If driver has been written using devLib commands, may not need to do anything
- OMS VME-58 driver required bit ordering modification, but no other changes
- Example library changes:

vxWorks	PCI/VME Bridge

`vxWorks.h`
`takLib.h`
`sysLib.h`
`intLib.h`
`iv.h`
`vxLib.h`
`sysLib.h`
	`devLib.h`
	`sys/types.h`

Victoria
The Place To Be

Migration of existing vxWorks driver

- Example function changes:

vxWorks	PCI/VME Bridge

`intConnect(...)`
`vxMemProbe(...)`
`sysBusToLocalAdrs(...)`
	`devConnectInterruptVME(...)`
	`devReadProbe(...)`
	`devMapAddr(...)`

- May need to make use of other EPICS OSI libraries
  - epicsMutex
  - epicsEvent

Driver development

Drivers known to work under PCI/VME bridge

Device
SIS3820 multiscaler
OMS VME-58 motor controller
DAC128V DAC
TVME200 IP carrier
SBS
IP330 ADC
IP-Unidig Digital IO

Drivers investigated but not yet converted

Device
quadEM electrometer

System performance

Performance

- Memory mapping I/O works
- Memory usage on host PC is not an issue so far
- Interrupt handling:
    - 1kHz uses approx 25% CPU on 3GHz Pentium
- Industry Pack interrupt processing is not yet stable
- Simple scan using motor, scaler, scan records operates well using interrupts

Benefits

- Ability to use VME hardware with Linux PC IOC
- Reuse of existing driver code with minimal modification
- Compatibility with existing EPICS records in synApps
- Do not need to pay for vxWorks
- Do not need to learn vxWorks
- PC host is easily upgraded
- Hardware costs are comparable

Further work

- Review signal handling architecture
- Fix up code structure
- Optimise code performance
- Produce documentation
- Make available to community
- Migrate additional drivers
- Resolve issue with Industry Pack interrupt handling

Acknowledgements

Russ Berg	Canadian Light Source
Glen Wright	Canadian Light Source
Mark Rivers	GSECARS
Ron Sluiter	APS
Tim Mooney	APS
Andrew Johnson	APS
Eric Norum	APS

The End

Questions?