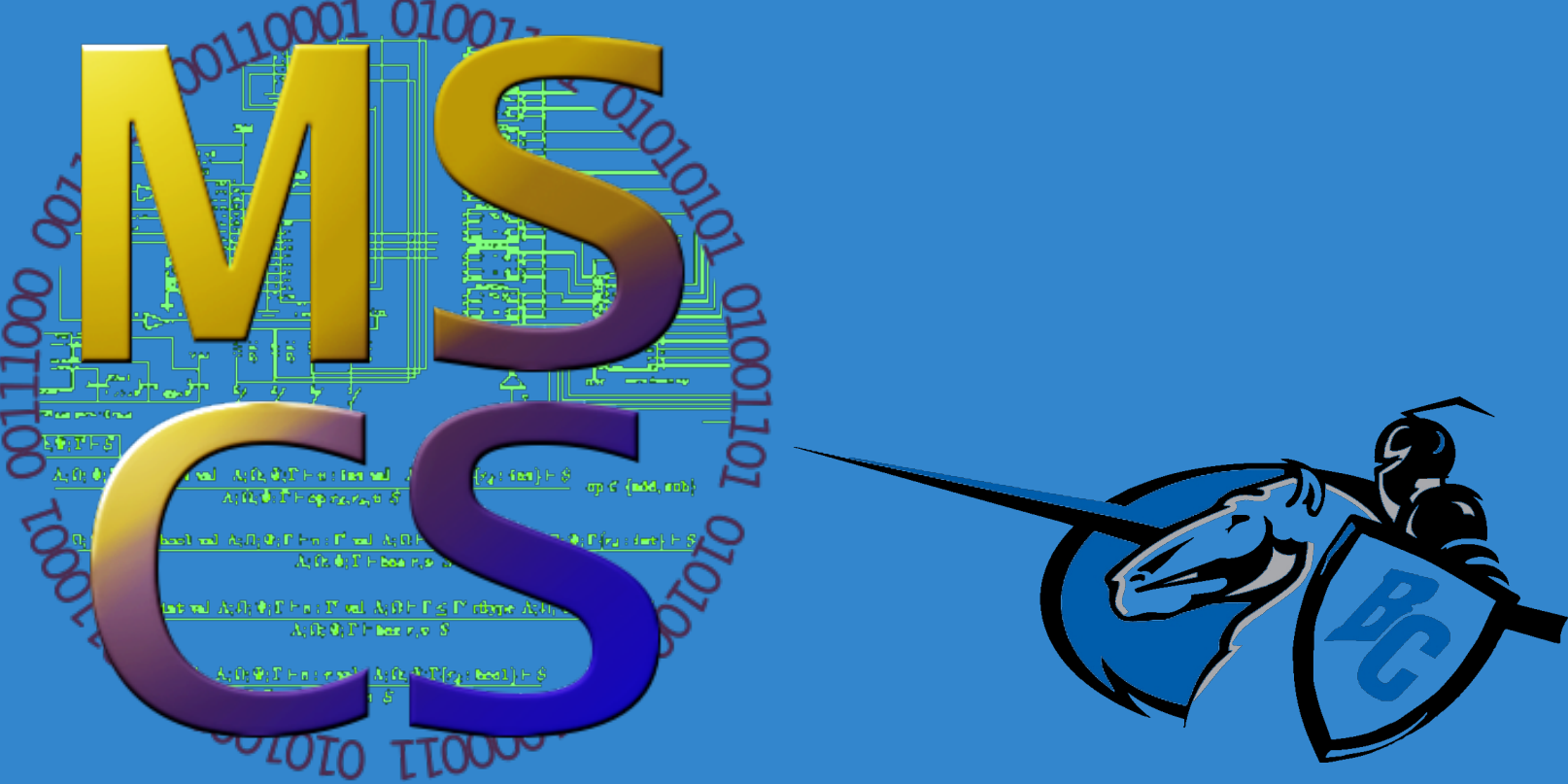


Embedded Xinu: Benefits of Teaching on a 32-bit Architecture

Rade Latinovich and Dr. Dennis Brylow



Introduction

- Embedded Xinu has only ran on 32-bit platforms previously
- Porting to a Raspberry Pi 3, a 64-bit platform
- Raspberry Pi 3 can run both 64-bit 32-bit code

64-bit Is Difficult

- Tried to initially use Raspberry Pi 3 in 64-bit mode
- Had to learn new instruction set, harder to understand compared to 32 bit instruction set
- Had to rewrite many files to comply with the 64 bit compiler
- Spent more time trying to understand new architecture than being productive
- Took less time to complete equivalent tasks in 32 bit
- Compatibility for old Raspberry Pi code was an attractive factor

Code Comparison

```
_vectors:
/* Current EL with SP0 */
b sync_addr /* Synchronous */
.balign 128
b irq_addr /* IRQ/vIRQ */
.balign 128
b fiq_addr /* FIQ/vFIQ */
.balign 128
b serr_addr /* SError/vSError */
/* Current EL with SPn */
b sync_addr /* Synchronous */
.balign 128
b irq_addr /* IRQ/vIRQ */
.balign 128
b fiq_addr /* FIQ/vFIQ */
.balign 128
b serr_addr /* SError/vSError */
/* Lower EL with Aarch64 */
b sync_addr /* Synchronous */
.balign 128
b irq_addr /* IRQ/vIRQ */
.balign 128
b fiq_addr /* FIQ/vFIQ */
.balign 128
b serr_addr /* SError/vSError */
/* Lower EL with Aarch32 */
b sync_addr /* Synchronous */
.balign 128
b irq_addr /* IRQ/vIRQ */
.balign 128
b fiq_addr /* FIQ/vFIQ */
.balign 128
b serr_addr /* SError/vSError */
```

Interrupt Vector Table for 64-bit ARM

```
/* ARM exception vector table. This is copied to VBAR register. */
/* Vector table has to be 32-byte aligned. */
.balign 0x20
_vectors:
ldr pc, reset_addr /* Reset handler */
ldr pc, undef_addr /* Undefined instruction handler */
ldr pc, swi_addr /* Software interrupt handler */
ldr pc, prefetch_addr /* Prefetch abort handler */
ldr pc, abort_addr /* Data abort handler */
ldr pc, reserved_addr /* Reserved */
ldr pc, irq_addr /* IRQ (Interrupt request) handler */
ldr pc, fiq_addr /* FIQ (Fast interrupt request) handler */
```

Interrupt Vector Table for 32-bit ARM

Meaning

- Example shows code that does equivalent tasks in 64 and 32 bit
- Code for 64 bit is longer to do the same task
- Same pattern can be seen in more places

Conclusions

- Students would have difficulty learning on a 64 bit platform
- Not ideal for teaching basic concepts
- Adds complexity where none is needed

Acknowledgements

I would like to acknowledge Dr. Dennis Brylow at Marquette University in the MSCS Department. This work was funded by NSF grant #CNS-1339392

References

- ARM Ltd. ARM Cortex-A53 MPCore Processor Technical Reference Manual, 2016
- ARM Ltd. ARMv8-A Architecture Reference Manual, 2017
- ARM Ltd. ARM Cortex-A Series Programmer's Guide for ARMv8-A, 2015
- Biggers, E., Harunani, F., Much, T., & Brylow, D. (2013, October). XinuPi: Porting a Lightweight Educational Operating System to the Raspberry Pi. In 2013 Proceedings of the Workshop on Embedded and Cyber-Physical Systems Education.