Simon Cook

simon.cook@embecosm.com

Ed Jones

ed.jones@embecosm.com

EMBECOSM®

Current in-tree architectures

Arch	Server	Desktop	Embedded	MCU	DSP	GPU	Hardware	Reg size
AArch64	✓	✓	✓				✓	64
AMDGPU						✓		32
ARM		✓	✓	✓			✓	32
BPF								64
Hexagon					✓			32
Lanai			?				?	32
Mips			✓				✓	32, 64
MSP430			✓	✓			✓	16
NVPTX						✓		16, 32, 64
PowerPC	✓	✓	✓				✓	32, 64
Sparc	✓	✓	✓				✓	64
SystemZ	✓						✓	32, 64
WebAsm								32, 64
X86	✓	✓	✓				✓	16, 32, 64
XCore			✓				✓	32
AVR			✓	✓			✓	8
RISC-V	✓	✓	✓				✓	32

Out-of-tree Architectures

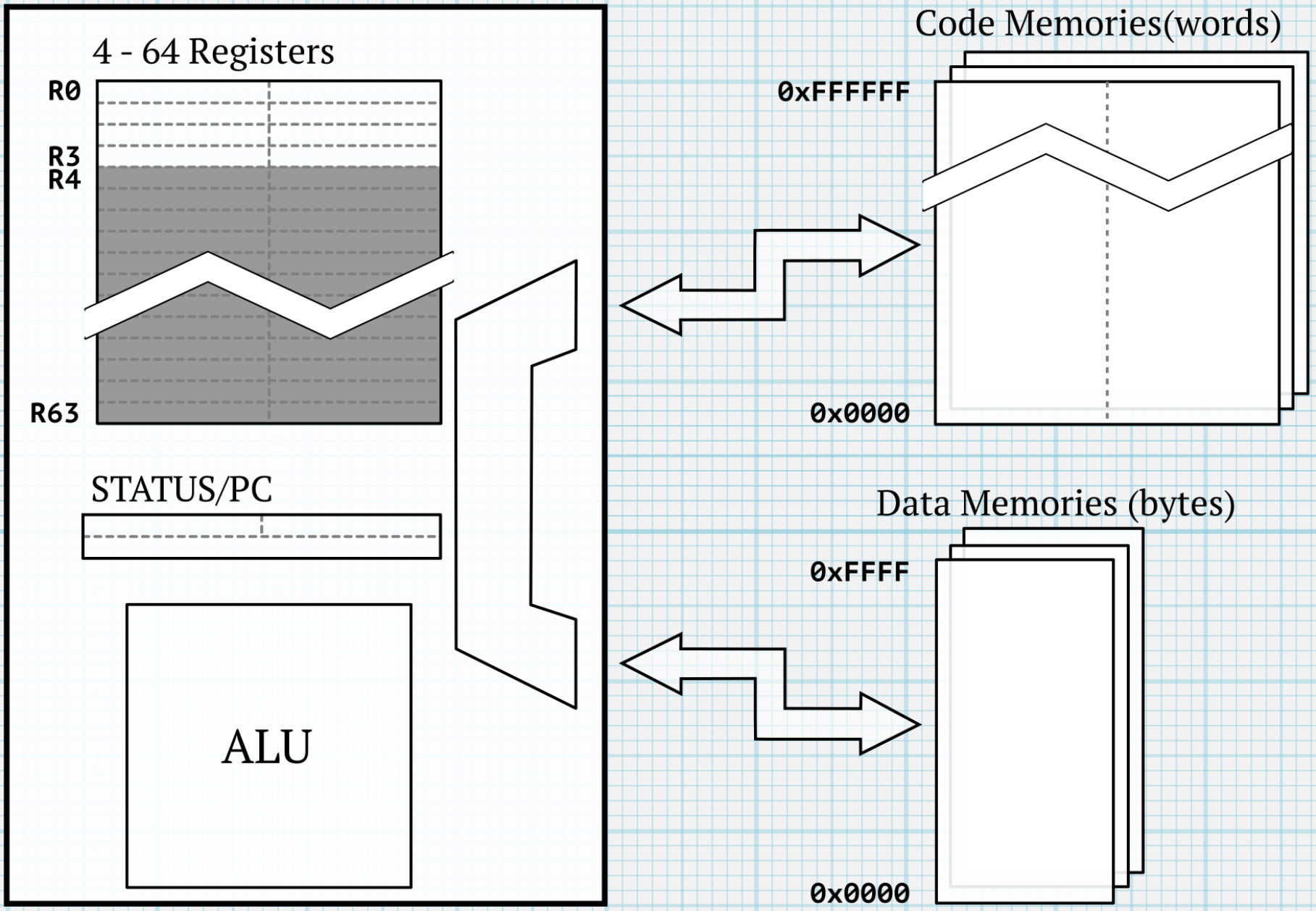
- MCUs, DSPs
- 8/16/24/40 bit registers, >8 bit bytes
- Novel addressing modes
- Complex register constraints
- Size > performance
- Can't always upstream
  - Exposes confidential hardware details
  - Only internal use
  - Lots of generic changes

Arch	Server	Desktop	Embedded	MCU	DSP	GPU	Hardware	Register size
AAP			✓	✓	✓		✓	16

- Community driven, incrementally update architecture to solve community needs
- Aim to be in-tree to avoid bit-rot
- Smooth the path for other back-ends to come in-tree
- RISC with MCU and DSP features

EMBECOSM®

AAP Design



AAP Design

AAP exists to improve LLVM. Some areas we're looking at:

- Non-octet chars
- Non-power of 2 registers
- Multiple function pointer sizes
- Finer-grained control over legalization
- Stack cheaper than registers

Non-octet Chars

Very common in DSPs, but 8-bit chars are hardcoded in a few places in LLVM/Clang

- `getPointerSizeInBits`
- `getTypeStoreSize/getTypeStoreSizeInBits`
- ....

Has appeared a couple of times in the mailing list (including /w patches). But patches never made it in.

Non-power of 2 registers

Also very common in DSPs. Generic LLVM only has power-of-2 in the MVT.

Support for common DSP register sizes would be nice (i24)

Again, patches have appeared on the mailing list.

Understandably, with no in-tree targets the patches weren't included.

Multiple function pointer sizes

Some architectures have code which lives in different address spaces.

Function pointers may be different sizes

- And different size pointers may point to the same function

Call and return instructions used may differ depending on pointer type

Can expand meaning of `address_space` in LLVM to work for functions.

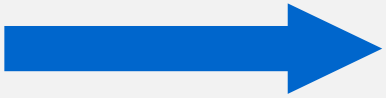
Finer-grained legalization control

- Some architectures have multiple register sizes
- May only be able to do a subset of operations on the larger register
- But if the larger register is legal, then LLVM will use it liberally.
- Can occur if your address registers are larger than your data registers.

Stack cheaper than registers

- Some architectures can perform operations on the stack directly.
- May be cheaper than registers:
  - For a subset of stack locations
  - For a subset of operations
- Want to *sometimes* treat stack slots like registers.

```
load    r0,   sp+2
add     r0,   r1
store   sp+2, r0
```



```
add     sp+2, r1
```

Stack cheaper than registers

- Cost of spilling a register varies
  - Free if subsequent uses of the value can operate directly on the stack
- Stack slot allocation varies
  - Allocate cheap stack slots to minimize number of loads into registers
- Instruction selection varies
  - Choose instructions to prefer operations which can use the stack

Differential Patches

D23665 – ELF Definitions
D23666 – Backend Stub
D23667 – Tablegen
D23771 – MC
D23772 – AsmParser
D23773 – InstPrinter

embecosm.com