CSE 1320 - Intermediate Programming

Bit Fields and Unions

Alex Dillhoff

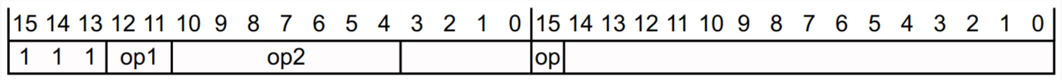
University of Texas at Arlington

Bit Fields

Bit fields in C allow members of a structure to be packed into a word of memory as part of the definition of the `struct` itself.

Bit Fields

Consider the 32-bit Thumb instruction encoding of the ARM v7 processor:



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	op1		op2											op															

The first 3 bytes are 1. We can use an *unnamed bit field* to pad our `struct` with 3 bytes.

Bit Fields

The operation bits **op1**, **op2**, and **op** will be represented as individual members.

The 4 bits between **op2** and **op** as well as the trailing 15 bits will be represented as reserved members.

Bit Fields

Since the entire instruction consists of 32 bits, we could represent this as an `int`.

With bit fields, we can create a 32-bit `struct` and separate the opcode from the reserved bits.

Bit Fields

```c
struct thumb_instr {
    unsigned int           : 3;
    unsigned int op1       : 2;
    unsigned int op2       : 7;
    unsigned int reserved1 : 4;
    unsigned int op        : 1;
    unsigned int reserved2 : 15;
};
```

Bit Fields

Example: thumb_instr.c

Bit Fields

Bit fields are useful in situations where resources are scarce, such as embedded systems.

Specifically, they're useful when the developer is trying to match some hardware specification.

Unions

C offers another feature that is useful for low resource environments: the `union`.

A `union` can hold multiple members, just like a `struct`.

Only one of the members of a `union` will be represented at any given time.

Unions

There are multiple practical uses for `union` instances.

Historically, a `union` was used so that multiple types could be represented without the additional memory overhead.

Unions

They can also be used for **type punning** – declaring multiple representations of the same memory space without any additional overhead.

Example: vec3f.c

Unions

Example: union_thumb_instr.c

Unions

There are some pitfalls to consider when using `union`s.

Memory that is dynamically allocated will be lost of another member is accessed.

Unions

Example: union_alloc.c