Managing Memory with Bit Fields

Philip Agaba
TEACHER

www.agabyte.com

Understanding Structure Padding

How is RAM Allocated to Structure Members?

When you create an object based on a structure, memory is *contiguously* allocated to the members of the structure.

```
struct jersey {

  char sponsor;
  char designer;
  int number;

} shirt;
```

```
struct jersey {

  char sponsor;
  char designer;
  int number;

} shirt;
```

What is the size of “shirt”?

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;
```

What is the size of “shirt”?

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

What is the size of “shirt”?

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // ~~6 bytes~~
```

What is the size of “shirt”?

Structure Padding

The CPU does not read one byte at a time

The CPU reads one “word” per-time

Word size

- 4 bytes (32 bit)
- 8 bytes (64 bit)

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

sponsor	designer	number	number
number	number		

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

sponsor	designer	number	number
number	number		

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 6 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  char designer;    // 1 byte
  int number;       // 4 bytes

} shirt;            // 8 bytes
```

```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // ???
```

```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // 1 +
```

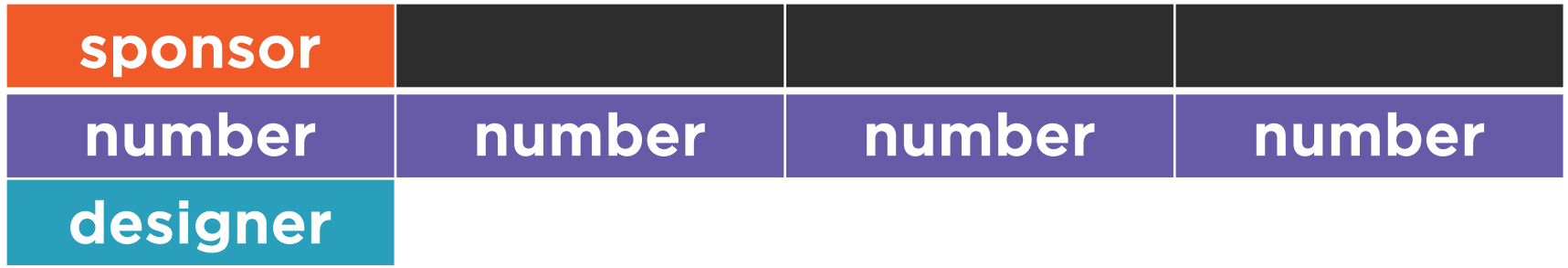
---

sponsor

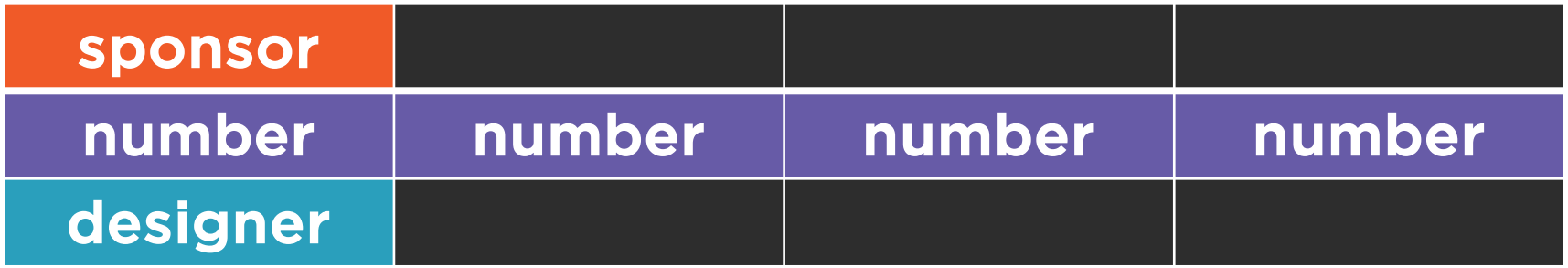
```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // 1 + 3 +
```

```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // 1 + 3 + 4 +
```

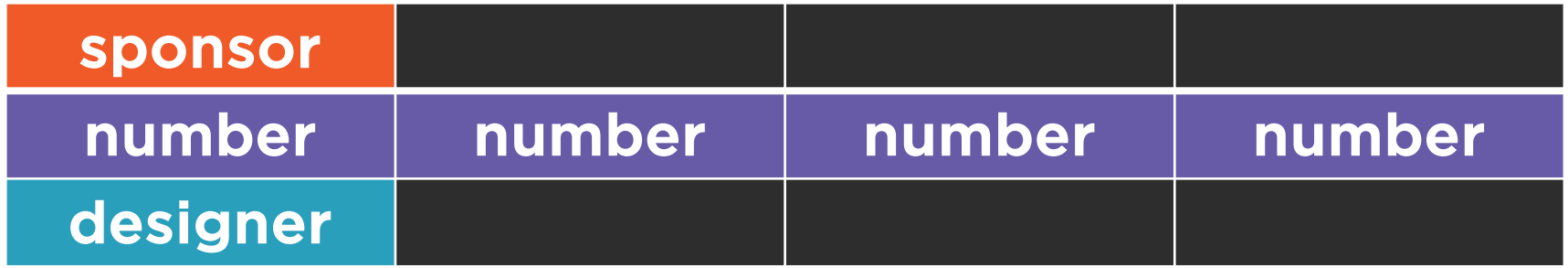
```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // 1 + 3 + 4 + 1 +
```



```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // 1 + 3 + 4 + 1 + 3
```



```
struct jersey {

  char sponsor;     // 1 byte
  int number;       // 4 bytes
  char designer;    // 1 byte

} shirt;            // 12 bytes!!!
```



A Bit Field

- Enables access to specific bits
- Makes it possible to work with data that's smaller than 8 bits
- Can help with memory optimization
- May help with hardware control
- Requires an integer data type

Bits	Max Value
1	1
2	2
3	7
4	15
5	31
6	63
7	127
8	255

Bits	Max Value
1	1
2	2
3	7
4	15
5	31
6	63
7	127
8	255

Module Summary

Now you know

- Why structures are padded
- How structures can be packed
- How to conserve memory with bit fields
- How to read and write raw data using bit fields within structures