

TASK 7

Embedded Systems Track

DATE :

6/5/2025

PRESENTED BY :

MOHAMED KADRY

Table Of Content

1. What is the purpose of typedef?
2. How are bit fields declared and what are their size limitations?
3. What happens if a bit field overflows?
4. How is typedef used with complex types like structs and unions?
5. What is the default underlying type of an enum?
6. How is a union different from a struct? When is using a union more memory-efficient?

1. What is the purpose of typedef?

Purpose: Creates a new name (alias) for an existing type.

Example:

```
typedef int Age;          // Age = int
Age myAge = 25;
```

2. How are bit fields declared, and what are their size limitations?

Declaration: Inside a struct, specify : N to allocate N bits.

Size Limit: Depends on the underlying type (e.g., unsigned int usually allows up to 32 bits).

Example:

```
struct {
    unsigned int enabled : 1; // 1-bit field
    unsigned int value   : 3; // 3-bit field (0-7)
} settings;
```

3. What happens if a bit field overflows?

Behavior: Truncates (cuts off extra bits).

Example:

```
struct {
    unsigned int num : 2; // 2 bits (0-3)
} x;
x.num = 6; // Truncates to 2 (binary 10)
```

4. How is typedef used with struct and union?

Example with struct:

```
typedef struct { // Struct type alias
    int x, y;
} Point;

typedef union { // Union type alias
    int num;
    char ch;
} Variant;
```

5. What is the default underlying type of an enum?

Default: int (but compiler-dependent).

Example:

```
enum { OFF, ON }; // OFF=0, ON=1 (ints)
```

6. How is a union different from a struct?

Difference:

- struct: Allocates space for all members (total size = sum of members).
- union: Shares memory for all members (size = largest member).

Example:

```
struct { int id; char code; }; // Allocates both
union { int id; char code; }; // Shares memory
```

7. When is using a union more memory-efficient?

Use Case: When only one member is needed at a time.

Example:

```
union {  
    int counter;  
    float percentage;  
} metric; // Stores either one at a time
```