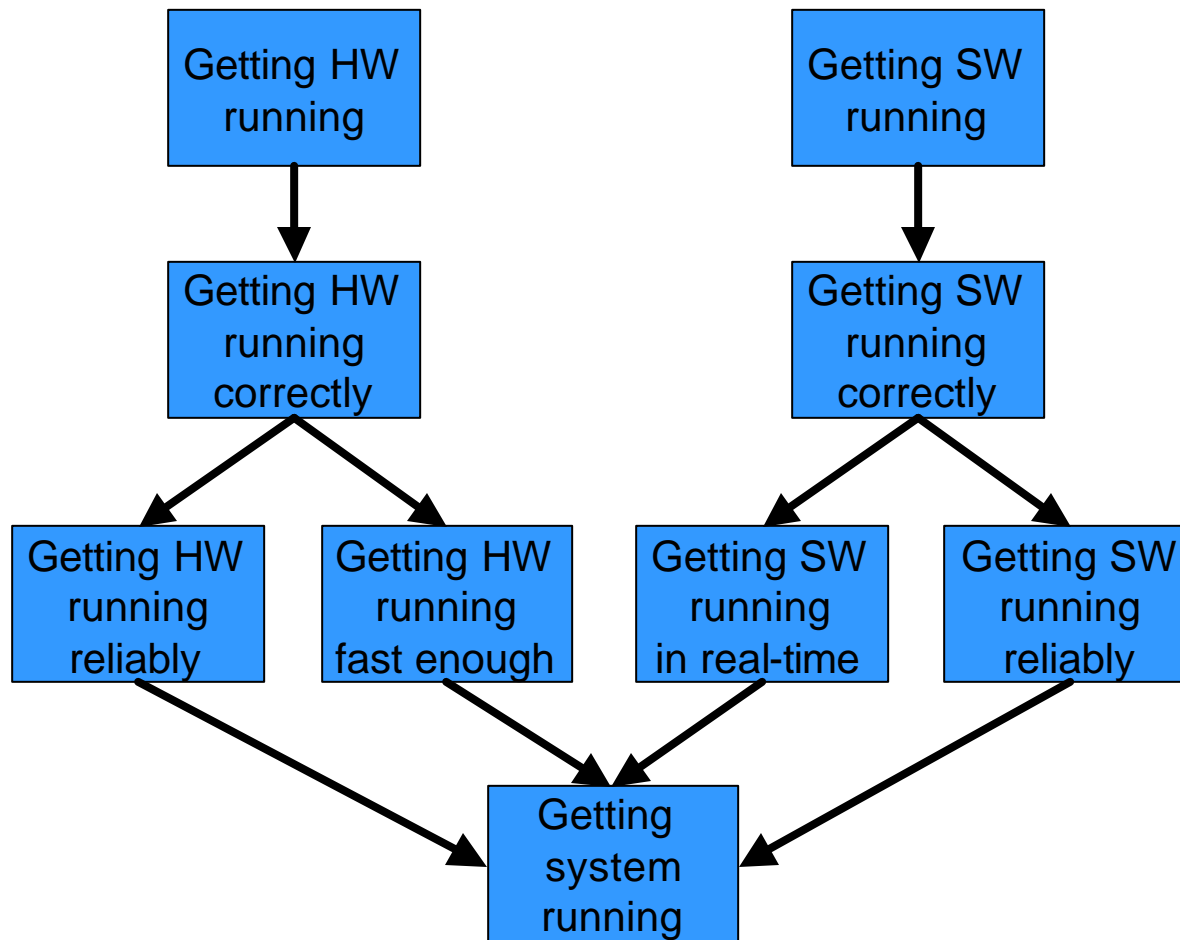


Architecture, Design and Implementation of Embedded Systems for Real-Time Applications

CpE-450 - Spring 05
Class 22

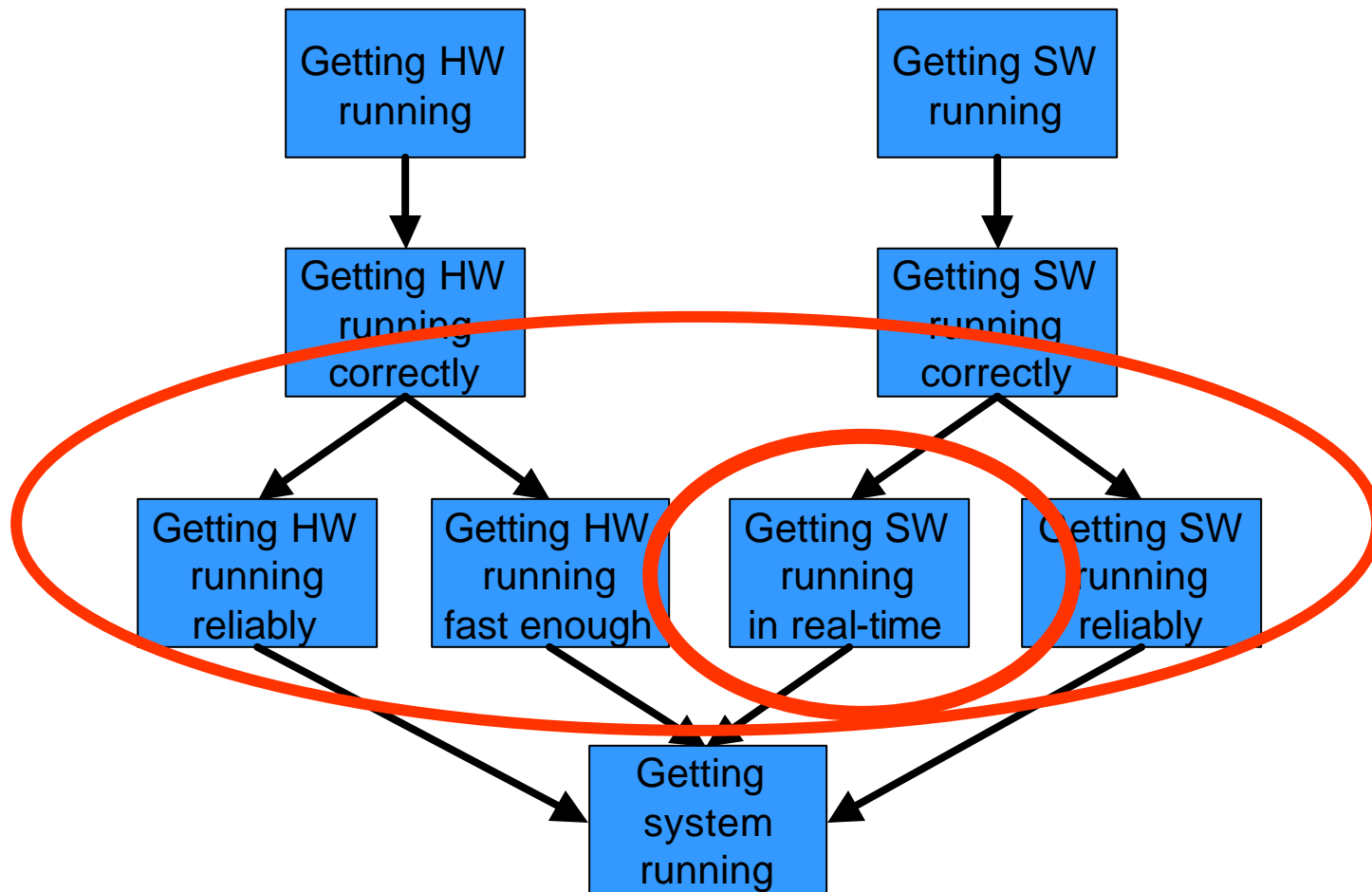
Bruce McNair
bmcnair@stevens.edu

Getting a Real-Time Embedded System Running



Getting a Real-Time Embedded System Running

- Evaluating performance and correctness



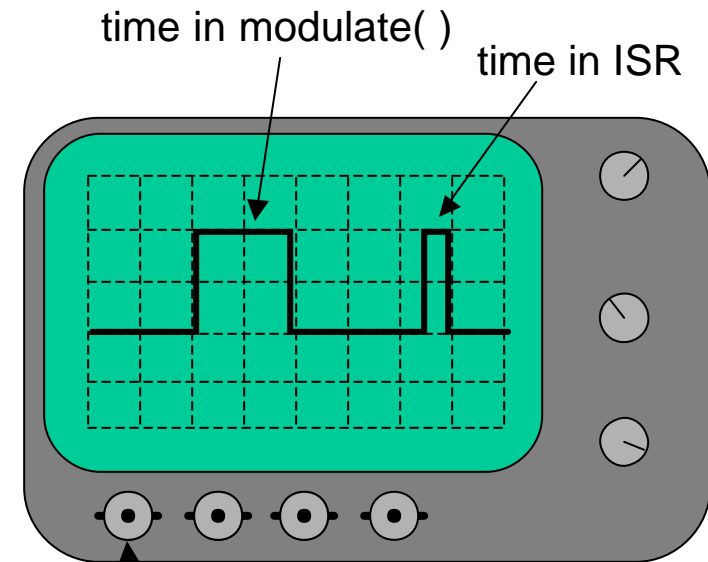
Real-time Observability

- Observing time spent in ISR, other functions

```
float sine_table[SIZE];
float cosine_table[SIZE];
struct signal
{
    float real;
    float imag;
};

void main( )
{
    boolean bits[BLOCK_LENGTH];
    signal sig[BLOCK_LENGTH];
    float mod[BLOCK_LENGTH];
    initialize_sine_table(*sine_table, *cosine_table);
    set_up_interrupts(INTERRUPT_ON_DATA_PRESENT);
    while(1)
    {
        get_data(bits);
        generate_baseband(bits, sig);
        set_output_flag(1);
        modulate(sig, mod, BLOCK_LENGTH, Fc, delta_t);
        set_output_flag(0);
        output_mod(mod);
        sleep( );
    }
}

void interrupt_handler_DATA_PRESENT(void)
{
    set_output_flag(1);
    /* wake up main( ) when data arrives */
    set_output_flag(0);
}
```



Real-time Tuning

- We now know how much time is spent in a routine
- We can tell how changes influence performance

Real-time Tuning

- We now know how much time is spent in a routine
 - We can tell how changes influence performance
- **Where do we spend our efforts improving performance?**

Real-time Performance

1. System is not running in real-time
2. System is running in real-time, but there isn't enough safety margin
3. System is running in real-time, but it is not yet complete – other features need to be added
4. System is running in real-time, but other features are likely to be added
5. System is running in real-time, but it would be desirable to improve performance allowing a slower (cheaper, lower power, ???) processor
6. .
7. .
8. .
9. Name your reason why you want to improve real-time performance.

Pareto Principle

- Economics:
 - 80% of the resources are consumed by 20% of the population

Pareto Principle

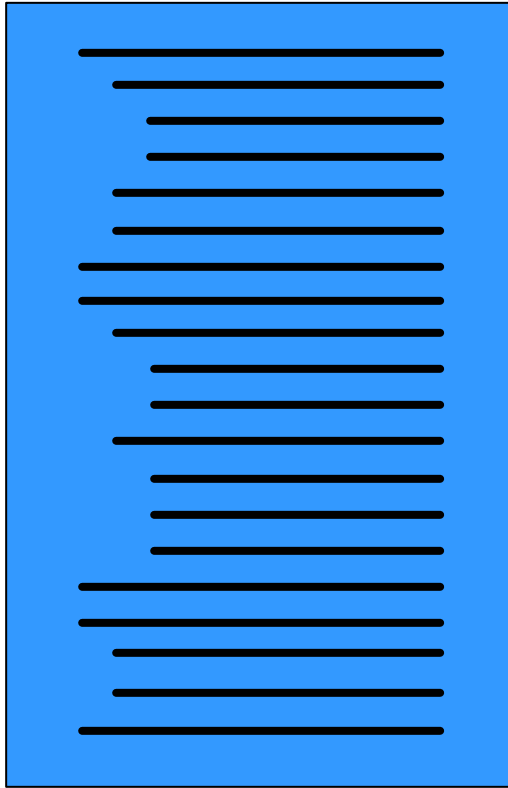
- Economics:
 - 80% of the resources are consumed by 20% of the population
- Reliability
 - 80% of the system failures are consumed by 20% of the components

Pareto Principle

- Economics:
 - 80% of the resources are consumed by 20% of the population
- Reliability
 - 80% of the system failures are consumed by 20% of the components
- Real-time programming
 - 80% of the real-time is consumed by 20% of the code

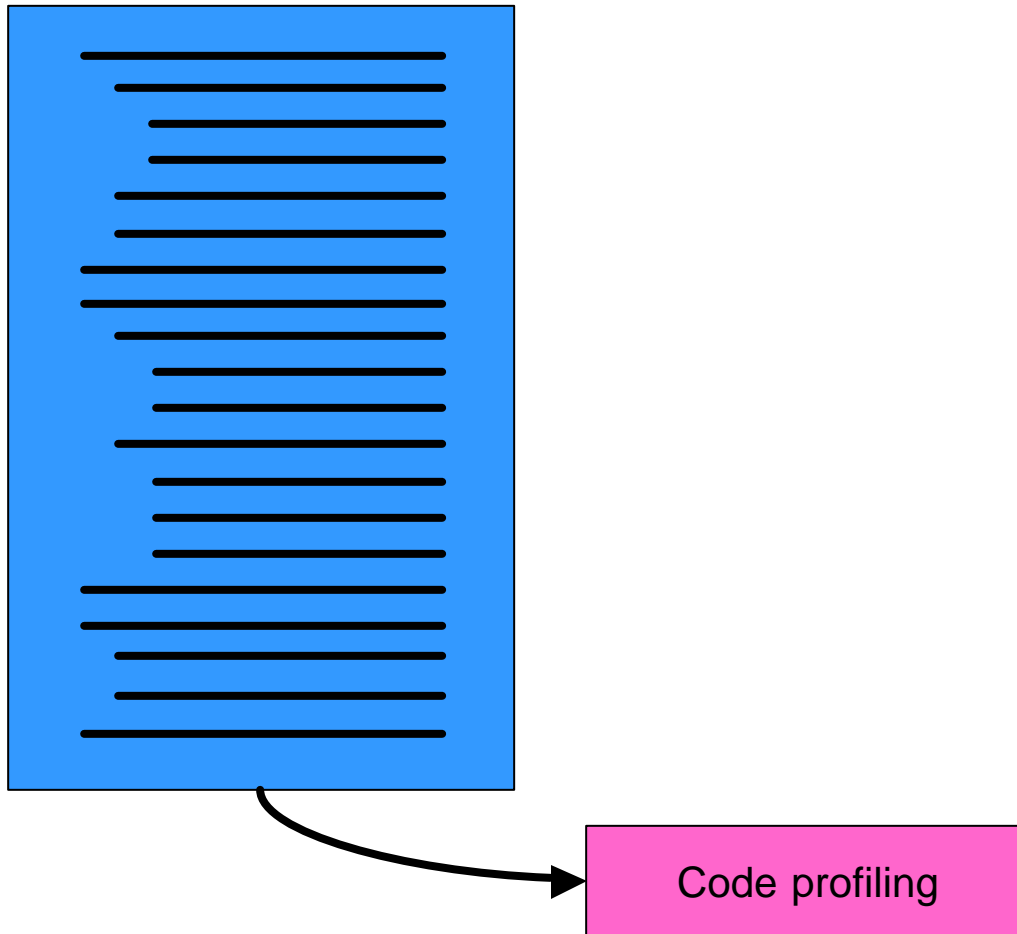
Applying Pareto Principle to Software

- Consider a generic software module:



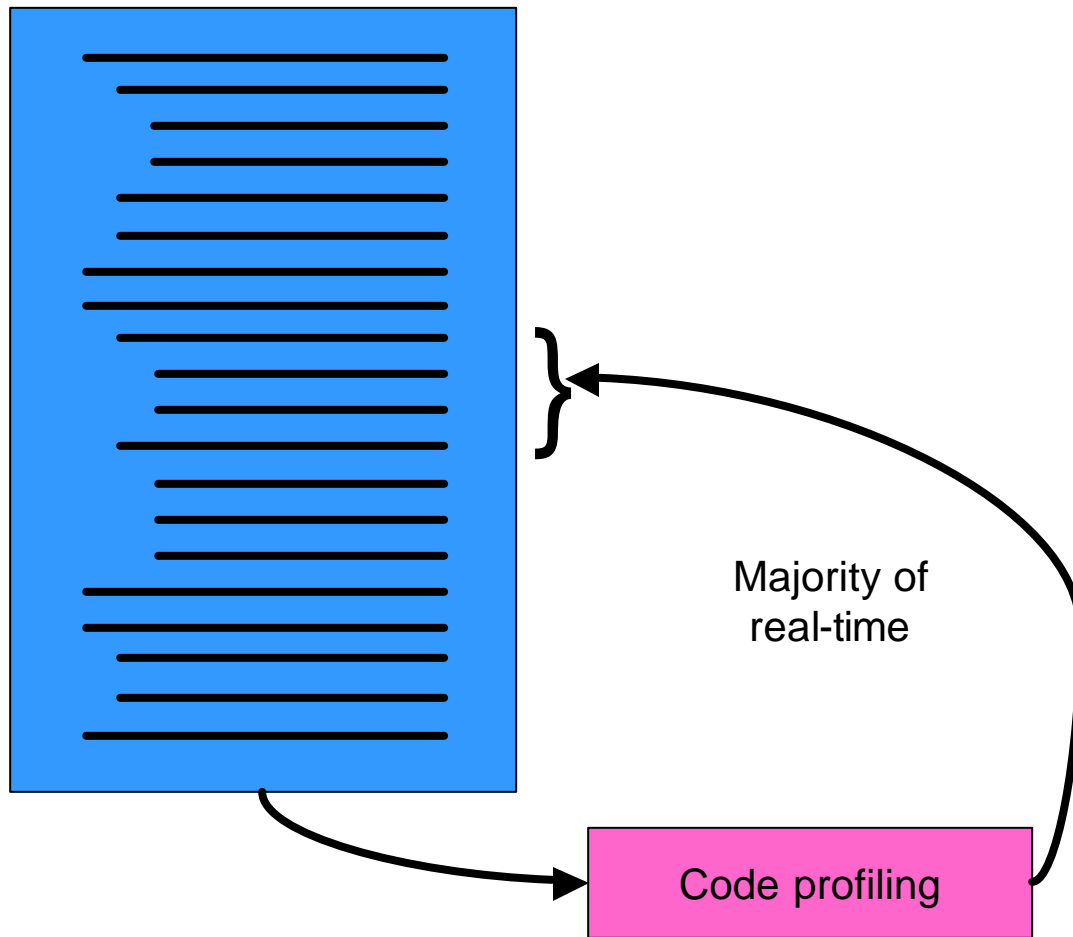
Applying Pareto Principle to Software

- Consider a generic software module:



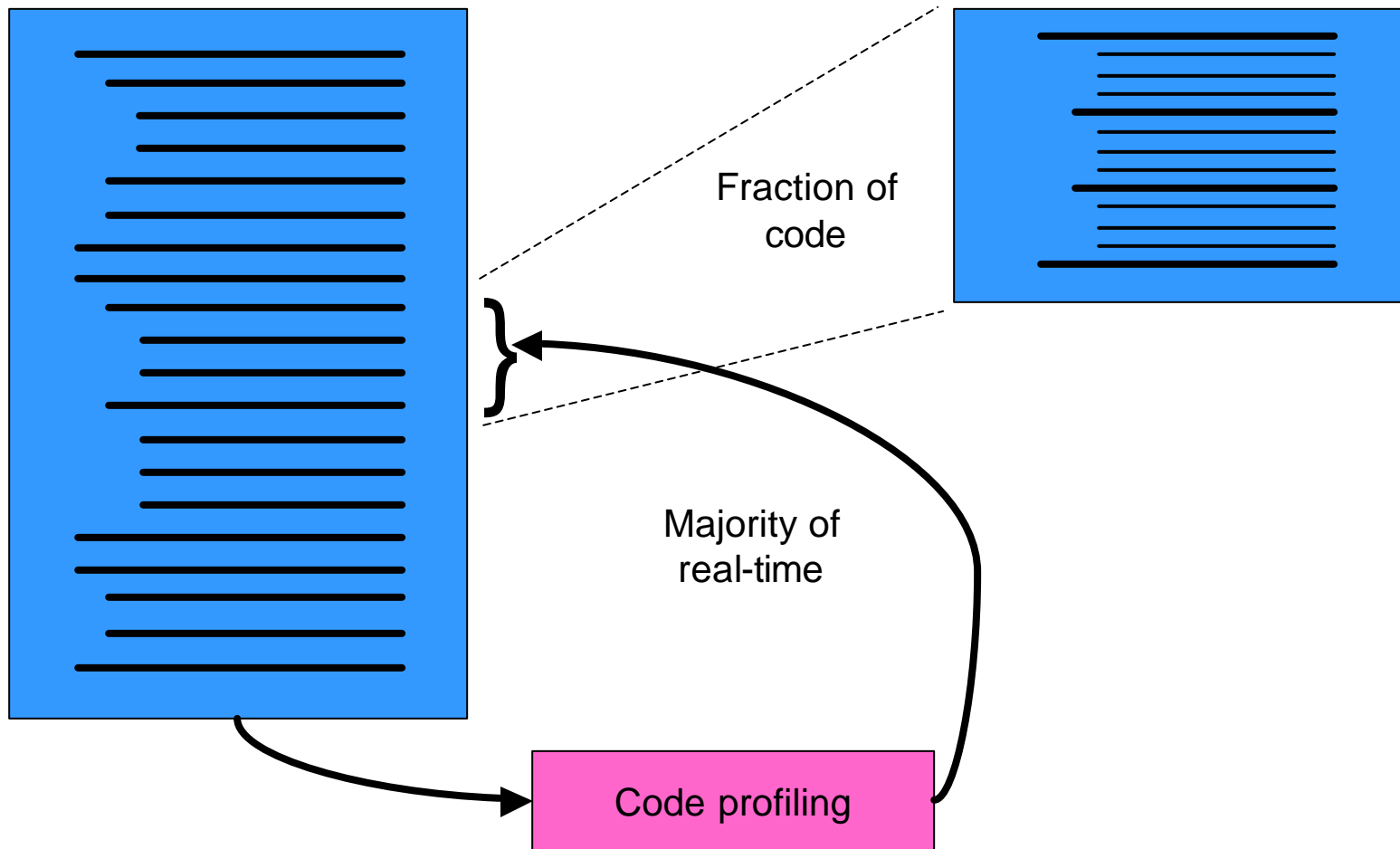
Applying Pareto Principle to Software

- Consider a generic software module:



Applying Pareto Principle to Software

- Consider a generic software module:



Continuing the Process

