

# EE445M/EE380L.12

## Embedded and Real-Time Systems/ Real-Time Operating Systems

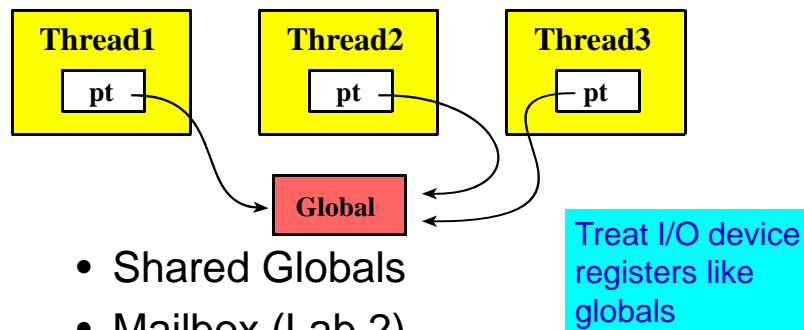
### Lecture 4: Thread Communication & Synchronization

Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

1

## Thread Communication/Sharing



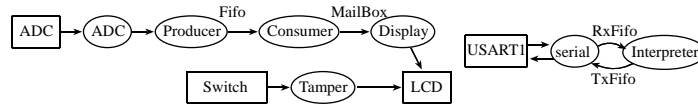
- Shared Globals
- Mailbox (Lab 2)
- FIFO queues (Lab 2)
- Message (Lab 6)

Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

2

# Thread Communication



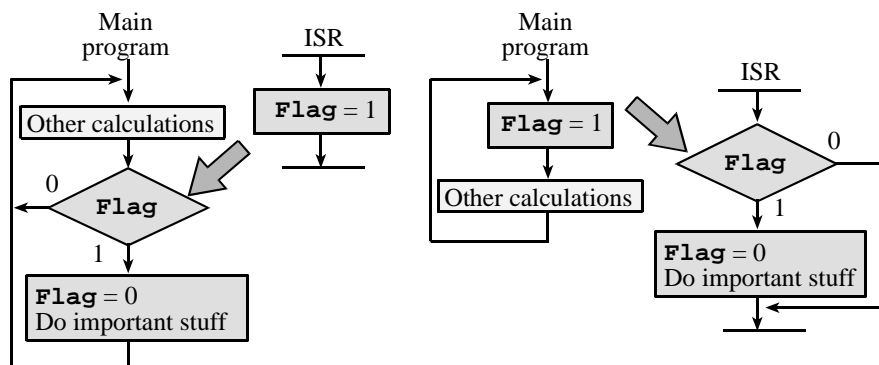
- Types
  - Data sharing (global variable)
  - Flag, Mailbox (one to one, unbuffered)
  - Pipes=FIFO (one to one, buffered, ordered)
  - Messages (many to many)
- Performance measures
  - Latency
  - Bandwidth
  - Error rate

Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

3

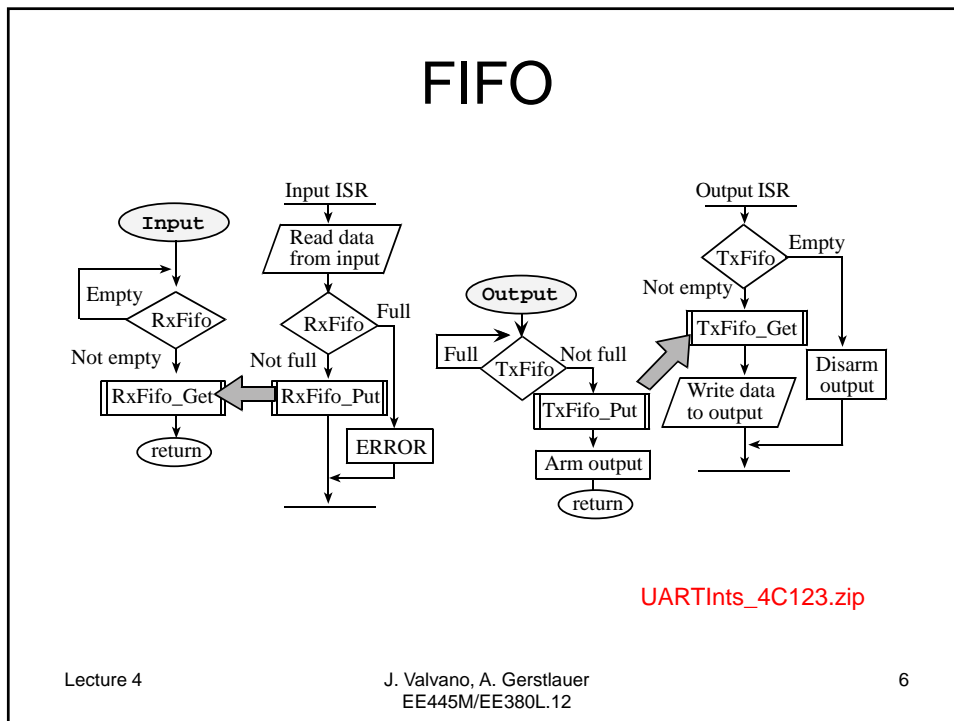
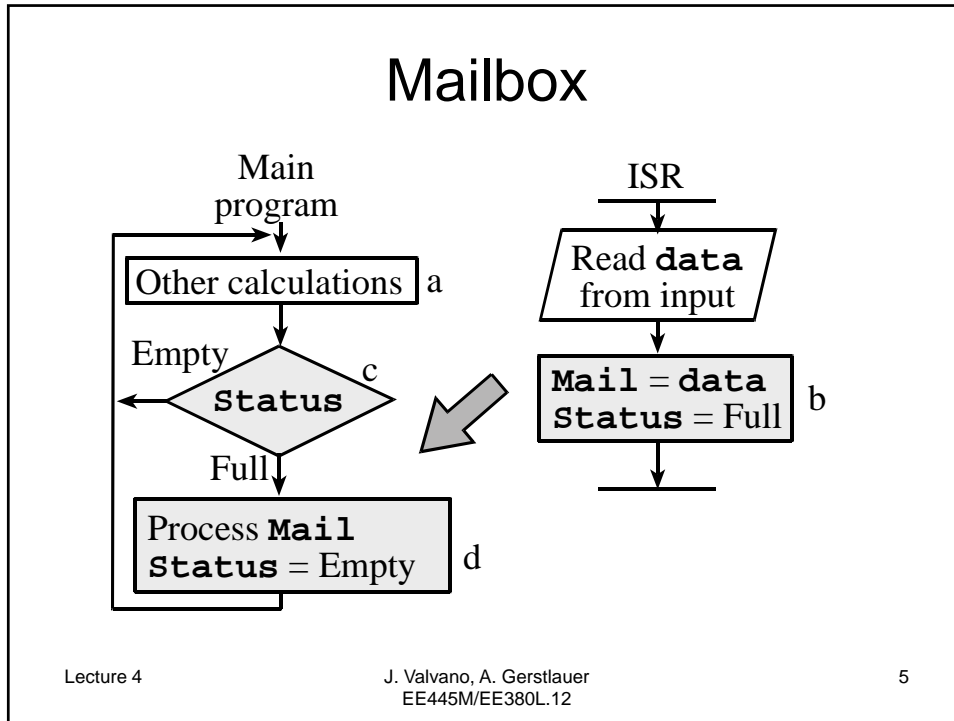
# Flag



Lecture 4

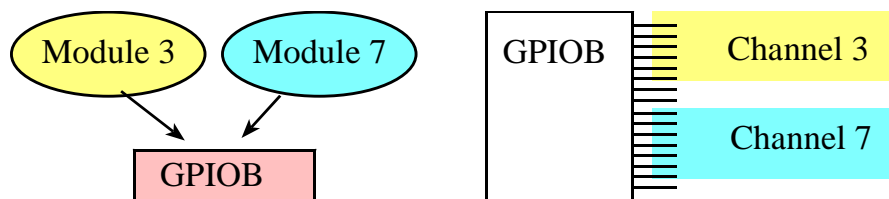
J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

4



## Race, Critical Section

- Two or more threads access the same global
  - Permanently allocated shared resource (memory, I/O port, ...)
- At least one access is a write



Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

7

## Race Condition

- Timing bug
  - Result depends on the sequence of threads
    - E.g. two threads writing to the same global
- Hard to debug
  - Depends on specific order/interleaving
    - Non-deterministic (external events)
    - Hard to reproduce/stabilize (“Heisenbug”)
- Critical or non-critical
  - Final program output affected?

Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

8

## Critical Section

- Load/store architecture
  - Write access changes official copy
  - Read access creates two copies
    - Original copy in memory
    - Temporary copy in register
- Non-atomic access sequence
  - Begins/ends with access to permanent resource
  - Involves at least one write
  - RMW(+W), WW(+R/W), WR(+W), RR(+W)

Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

9

## Thread-Safe, Reentrant

- Thread-safe code
  - No global resources
    - Variables in registers, stack
  - No critical section
    - No write access sequence
  - Mutual exclusion
    - Make accesses atomic (no preemption)
    - Prevent other threads from entering critical section
- Reentrant code
  - Multiple threads can (re-)enter same section
    - No non-atomic RMW, WW, WR sequence

Lecture 4

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

10

## Mutual Exclusion

- Disable all interrupts
  - Make atomic
- Lock the scheduler
  - No other foreground threads can run
  - Background ISR will occur
- Mutex semaphore
  - Blocks other threads trying to access info
  - All nonrelated operations not delayed
    - Thread-safe, but not reentrant

Measure time with interrupts disabled  
 - Maximum time  
 - Total time

Lecture 4

**LDREX**  
**STREX** Cortex-M3/M4F Instruction Set, pg. 50

## Thread Synchronization

- Sequential
- Rendezvous, Barrier
  - Fork/spawn & join
- Trigger, event flags
  - OR, AND
  - I/O event (e.g., I/O edge, RX, TX)
- Time
  - Periodic time triggered (e.g., TATOMIS)
  - Sleep

Lecture 4

J. Valvano, A. Gerstlauer  
 EE445M/EE380L.12

12