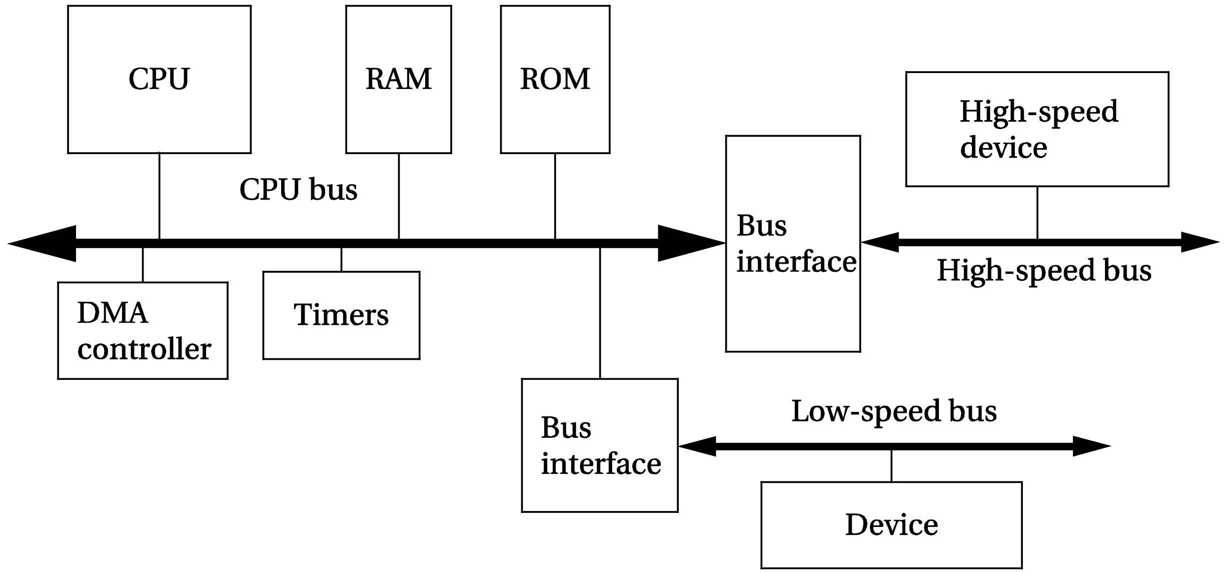
Computing Platforms

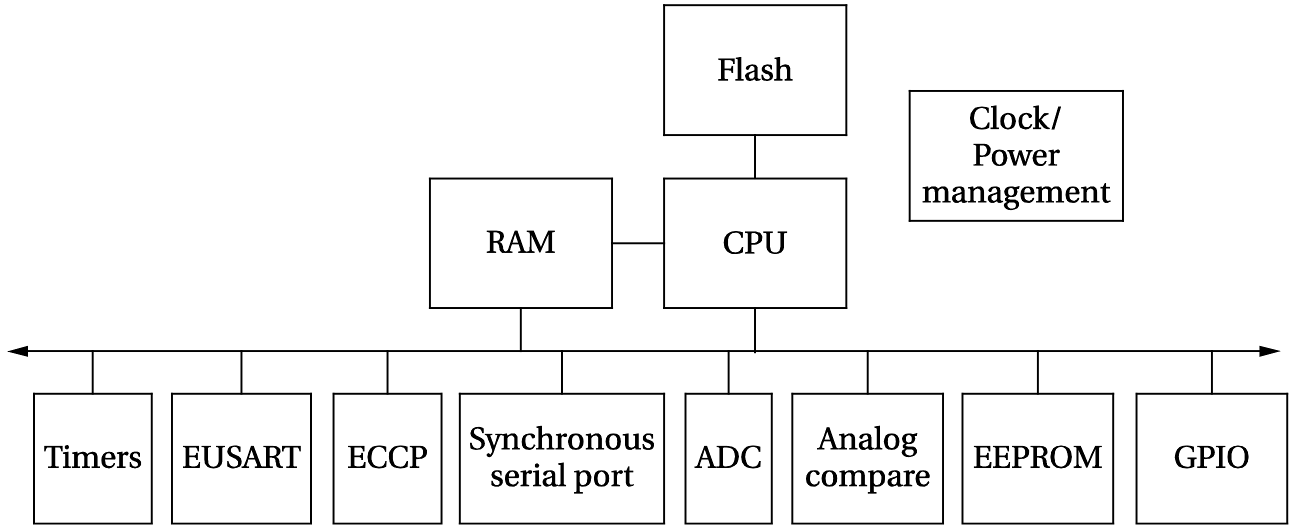
- Platform organization.
- Busses.
- Memory devices.
- I/O devices.

Computing platform architecture



- DMA provides direct memory access.
- Timers used by OS, devices.
- Multiple busses connect CPU, memory to devices.

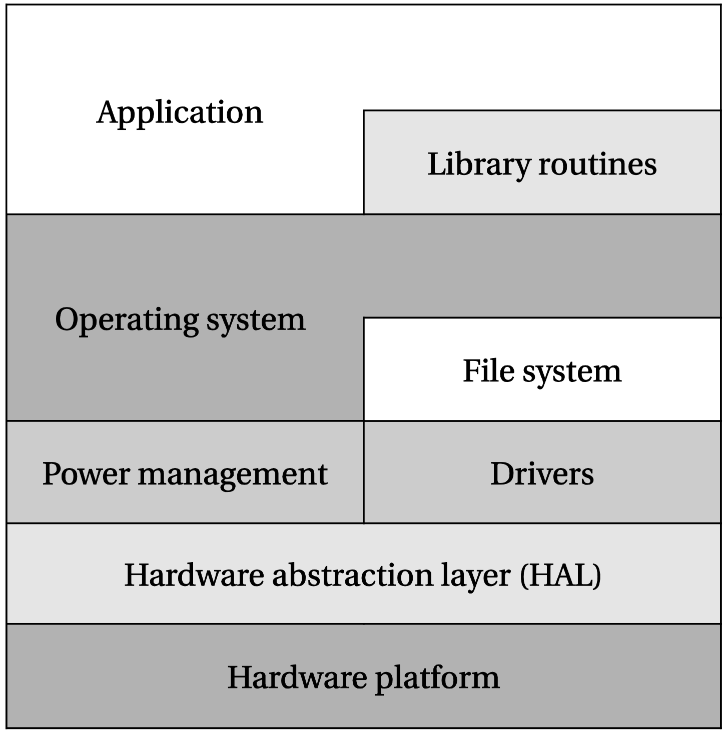
PIC16F882



- Harvard architecture---flash memory separately programmed.
- Multiple I/O devices.

Platform software

- Platform software provides core functions, utilities.
- Low-level functions depend on architecture---interrupt vectors, etc.

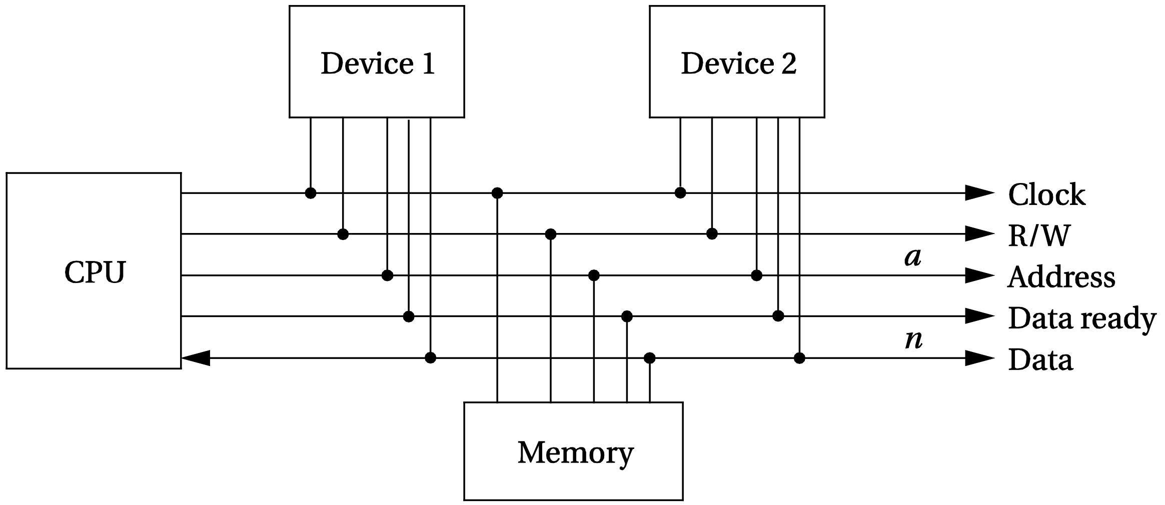


CPU buses

- Bus allows CPU, memory, devices to communicate.
    - Shared communication medium.
- A bus is:
    - A set of wires.
    - A communications protocol.

Microprocessor busses

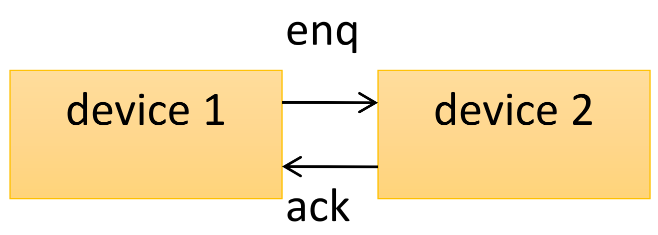
- Clock provides synchronization.
- R/W is true when reading (R/W’ is false when reading).
- Address is a-bit bundle of address lines.
- Data is n-bit bundle of data lines.
- Data ready signals when n-bit data is ready.

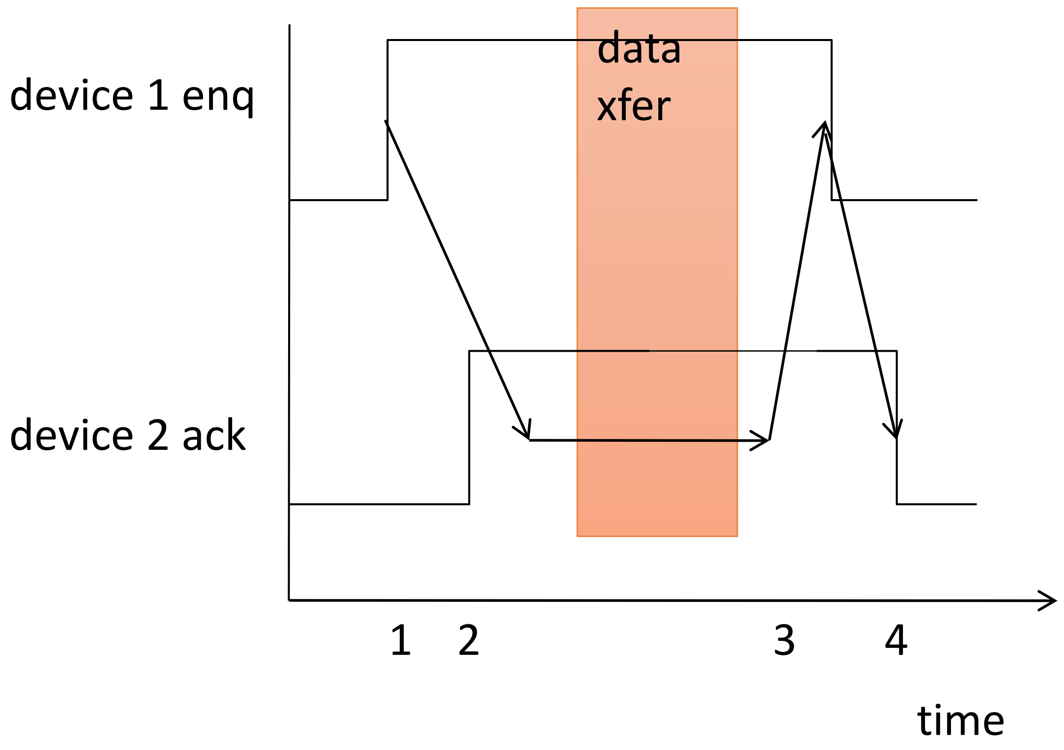


Bus protocols

- Bus protocol determines how devices communicate.
- Devices on the bus go through sequences of states.
  - Protocols are specified by state machines, one state machine per actor in the protocol.
- May contain asynchronous logic behavior.

Four-cycle handshake

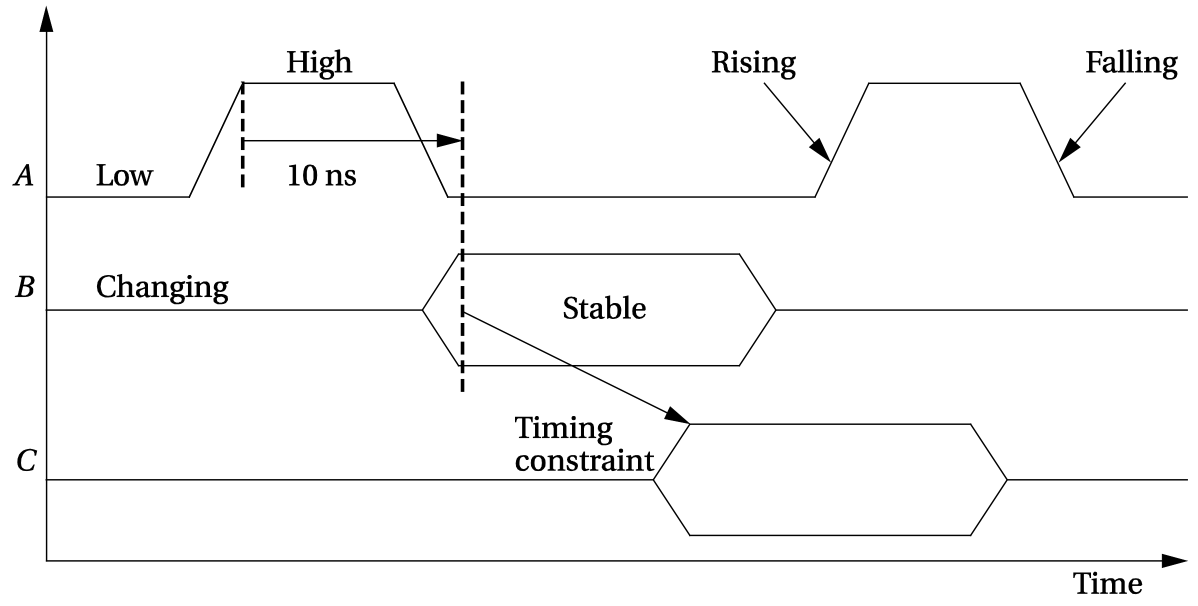




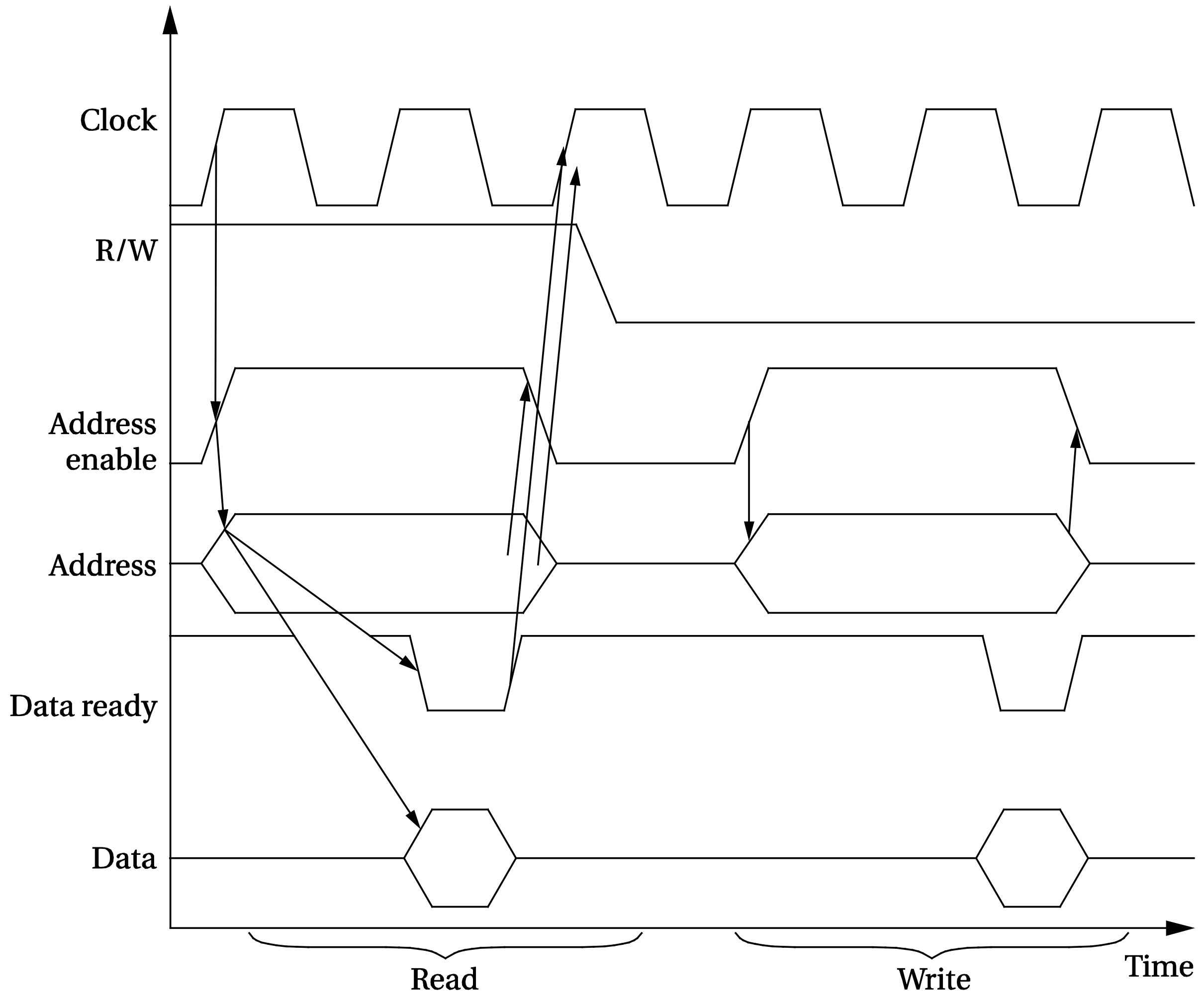
Four-cycle handshake, cont’d.

1. Device 1 raises enq.
2. Device 2 responds with ack.
3. Device 2 lowers ack once it has finished.
4. Device 1 lowers enq.

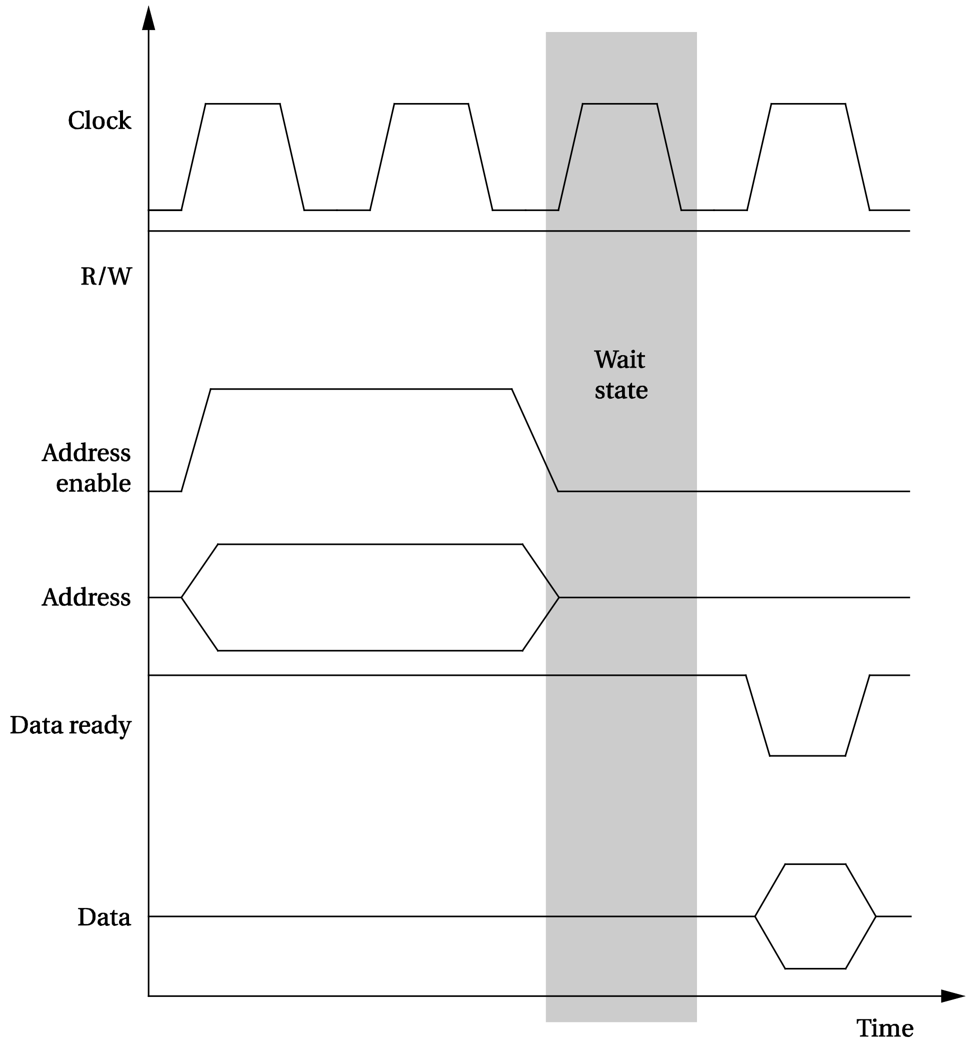
Timing diagrams



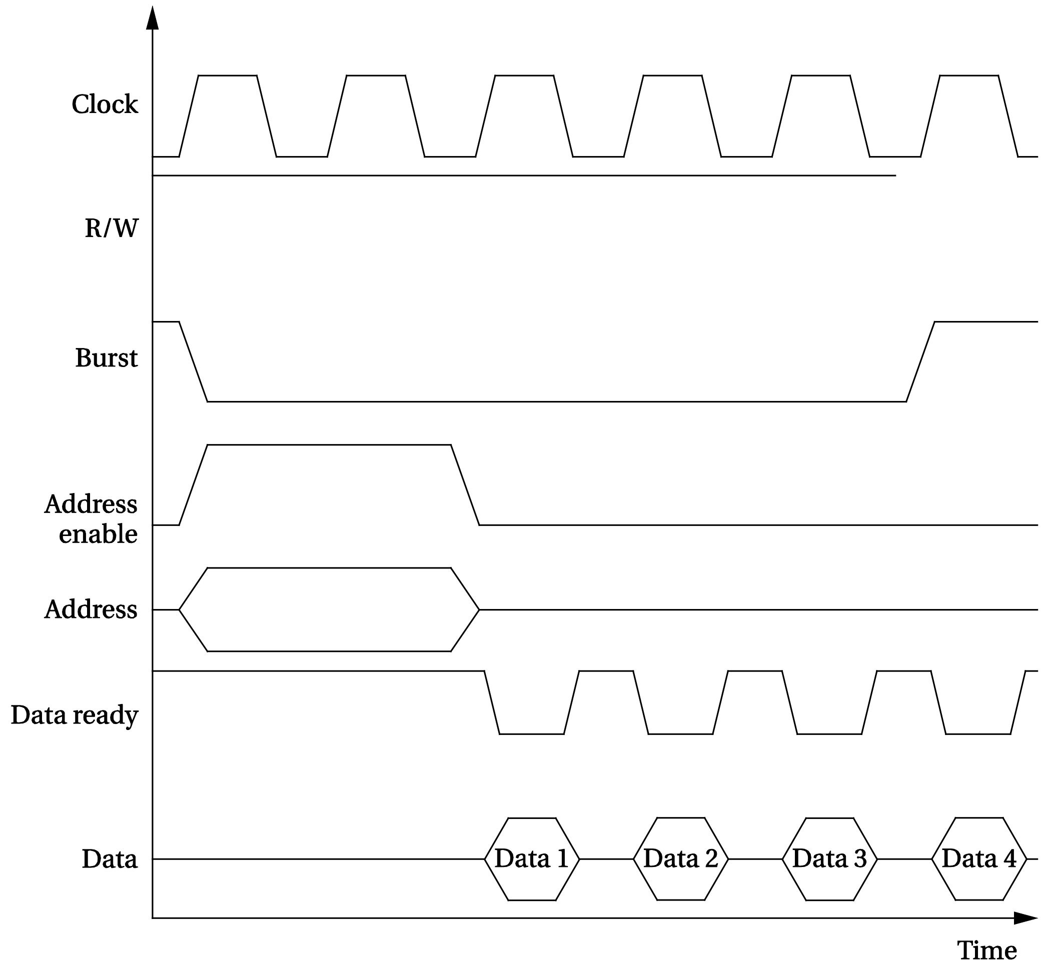
Bus read


Clock
R/W
Address enable
Address
Data ready
Data
Read
Write
Time

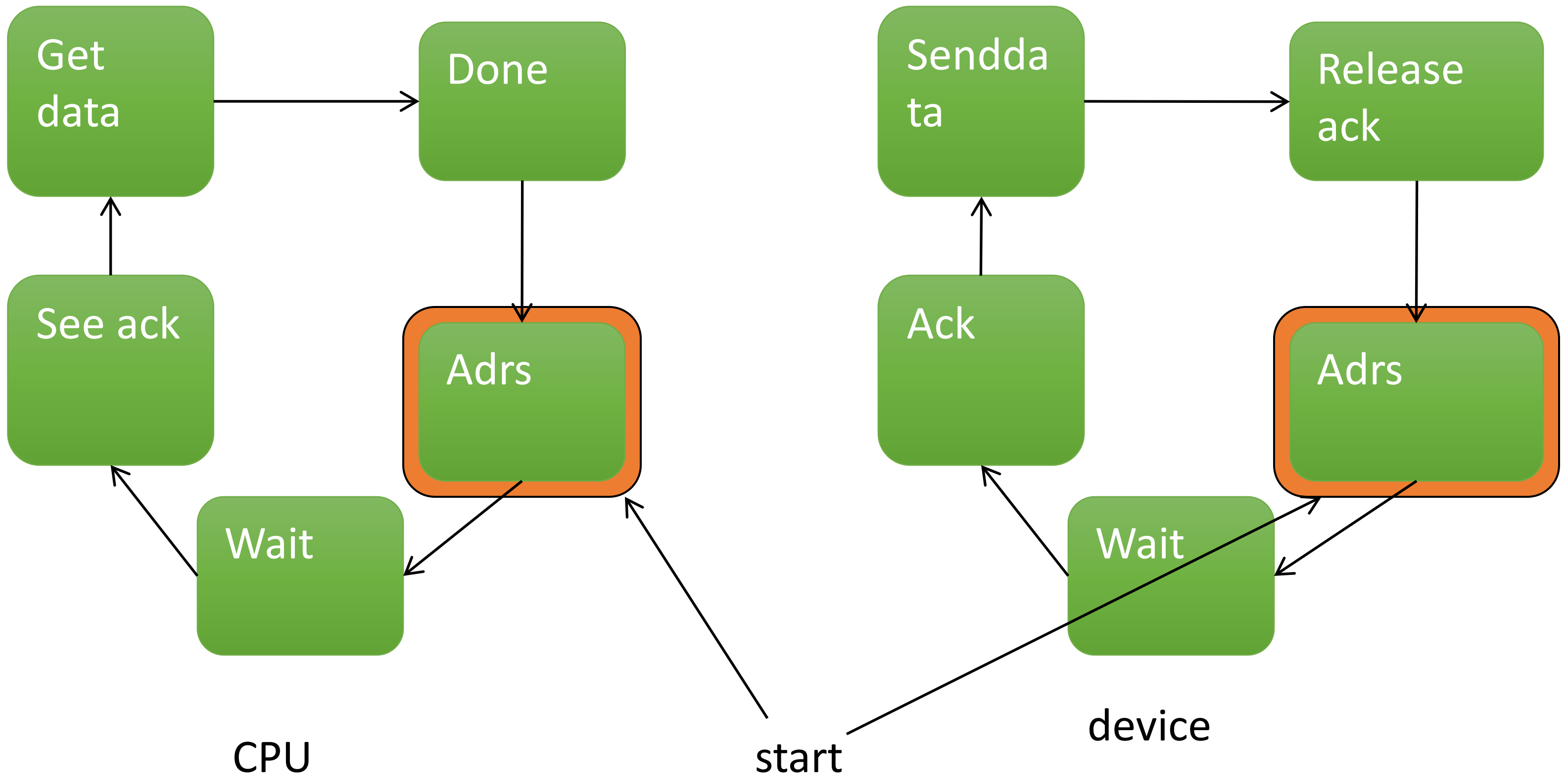
Bus wait state



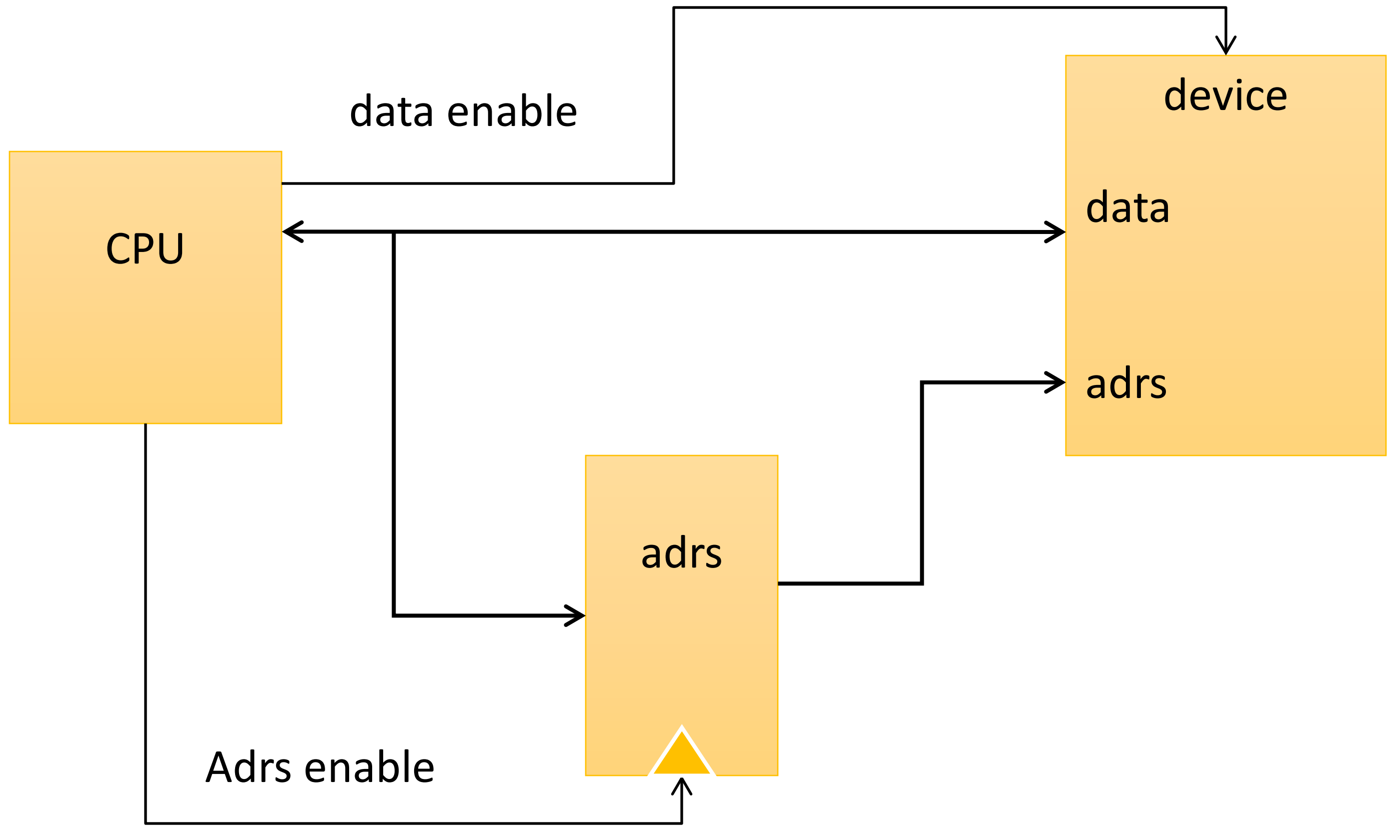
Bus burst read



State diagrams for bus read

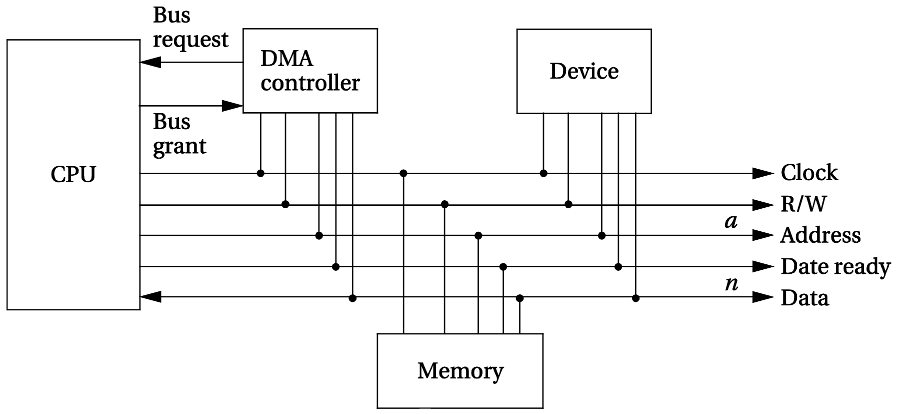


Bus multiplexing



DMA

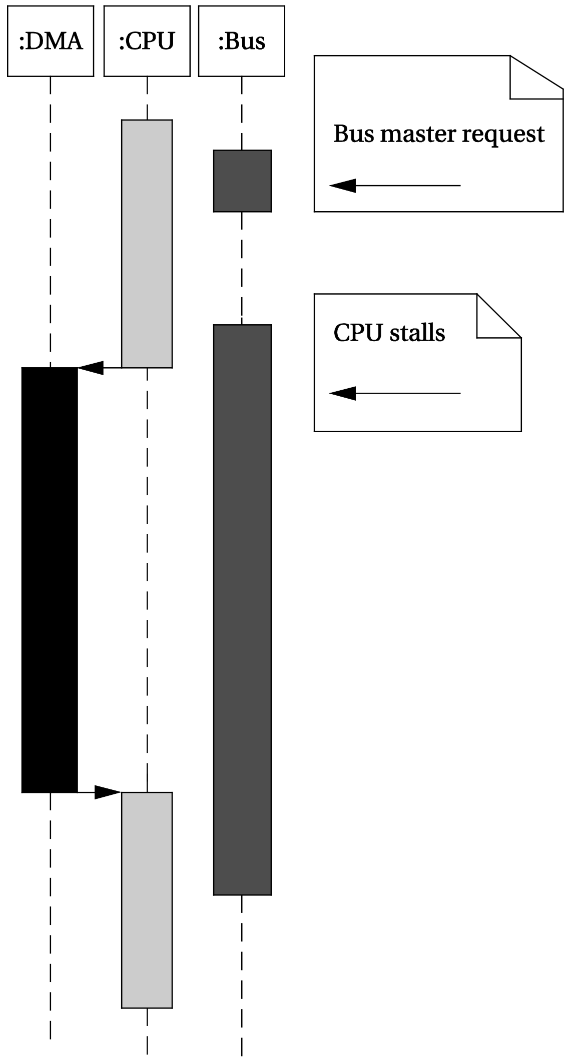
- Direct memory access (DMA) performs data transfers without executing instructions.
  - CPU sets up transfer.
  - DMA engine fetches, writes.
- DMA controller is a separate unit.



Bus mastership

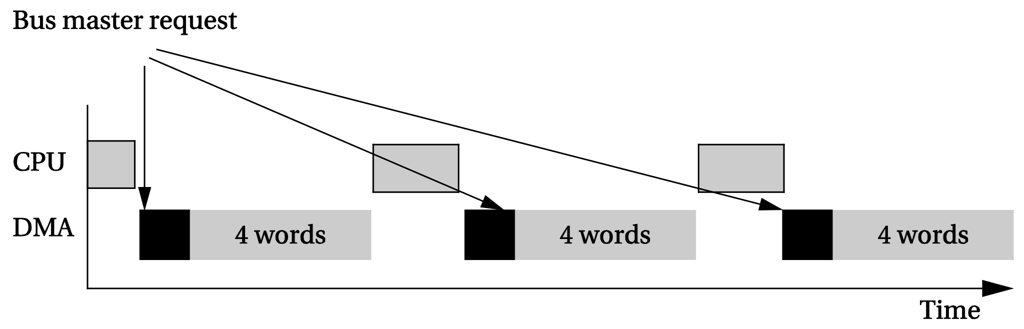
- By default, CPU is bus master and initiates transfers.
- DMA must become bus master to perform its work.
  - CPU can’t use bus while DMA operates.
- Bus mastership protocol:
  - Bus request.
  - Bus grant.

Bus transfer sequence diagram



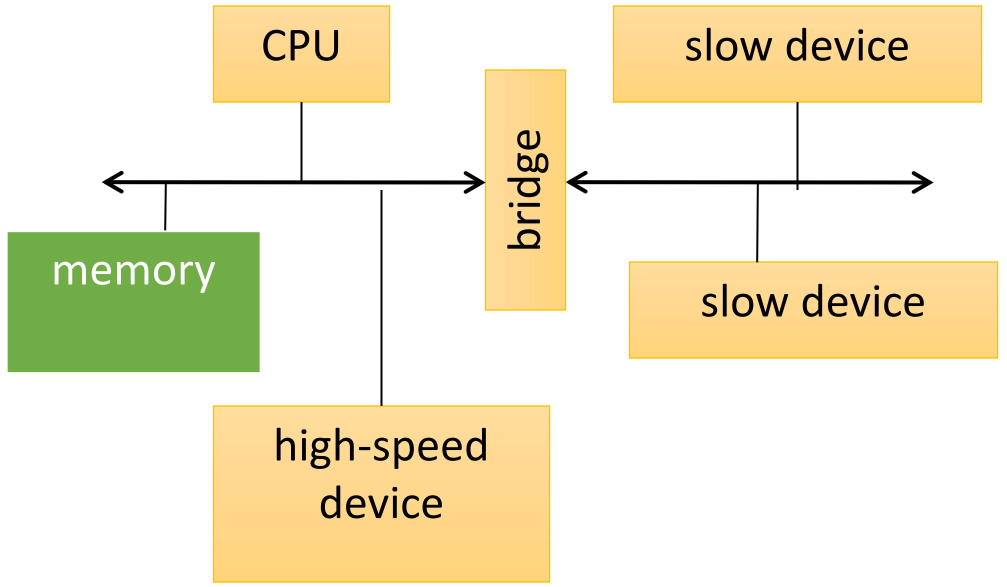
DMA operation

- CPU sets DMA registers for start address, length.
- DMA status register controls the unit.
- Once DMA is bus master, it transfers automatically.
  - May run continuously until complete.
  - May use every n^{th} bus cycle.

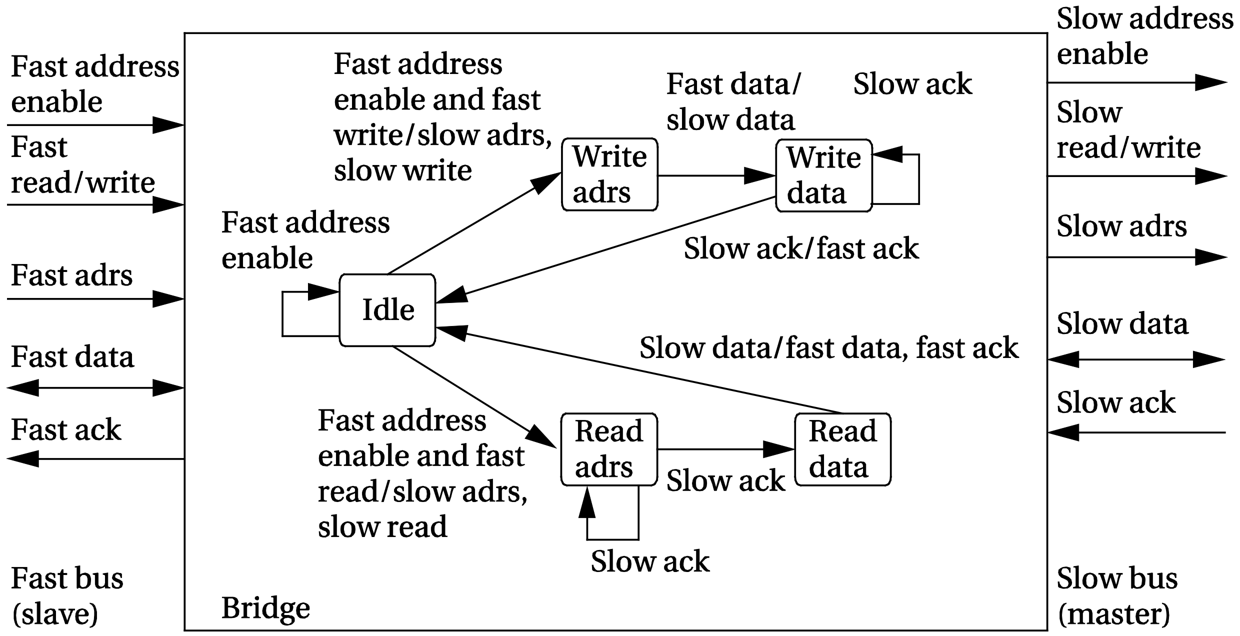


System bus configurations

- Multiple busses allow parallelism:
  - Slow devices on one bus.
  - Fast devices on separate bus.
- A bridge connects two busses.

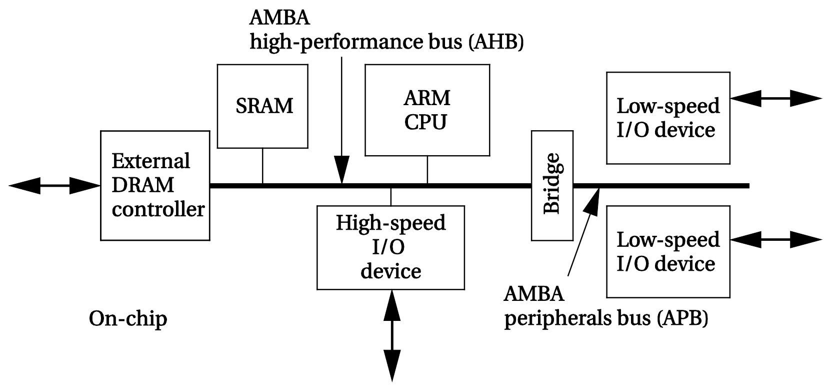


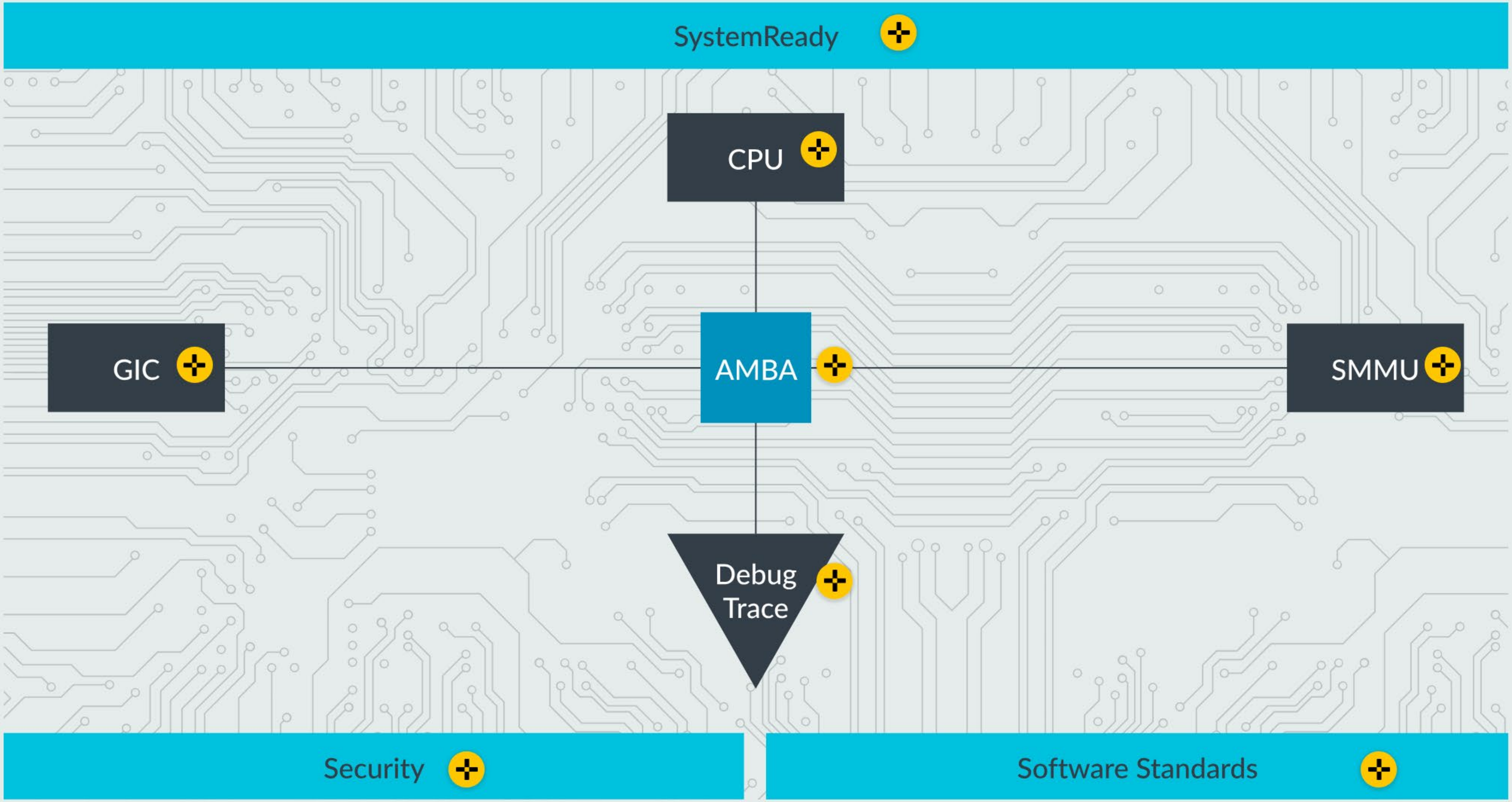
Bridge state diagram

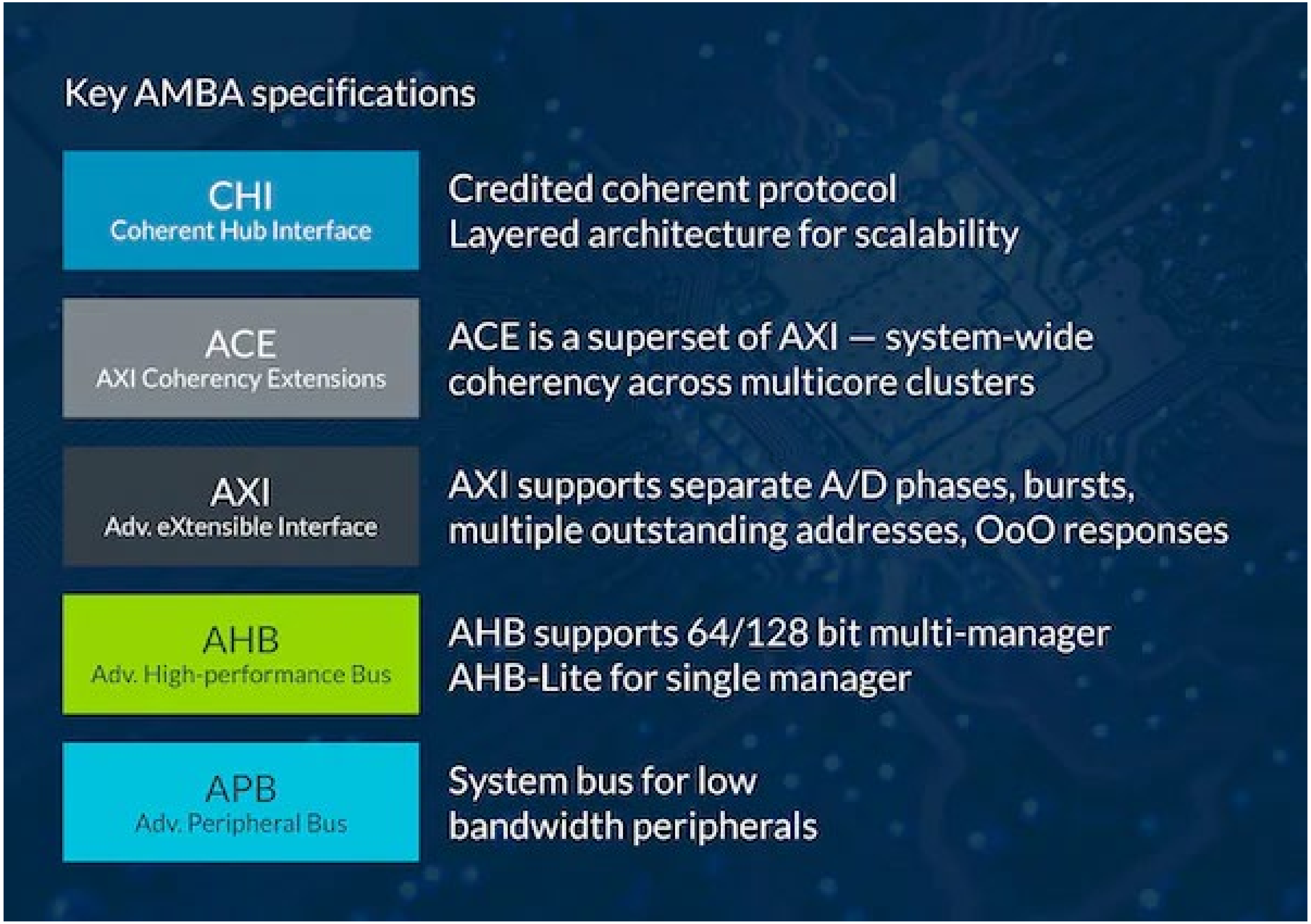


ARM AMBA bus

- Two varieties:
    - AHB is high-performance.
    - APB is lower-speed, lower cost.
- AHB supports pipelining, burst transfers, split transactions, multiple bus masters.
- All devices are slaves on APB.




SystemReady
CPU
GIC
AMBA
SMMU
Debug Trace
Security
Software Standards

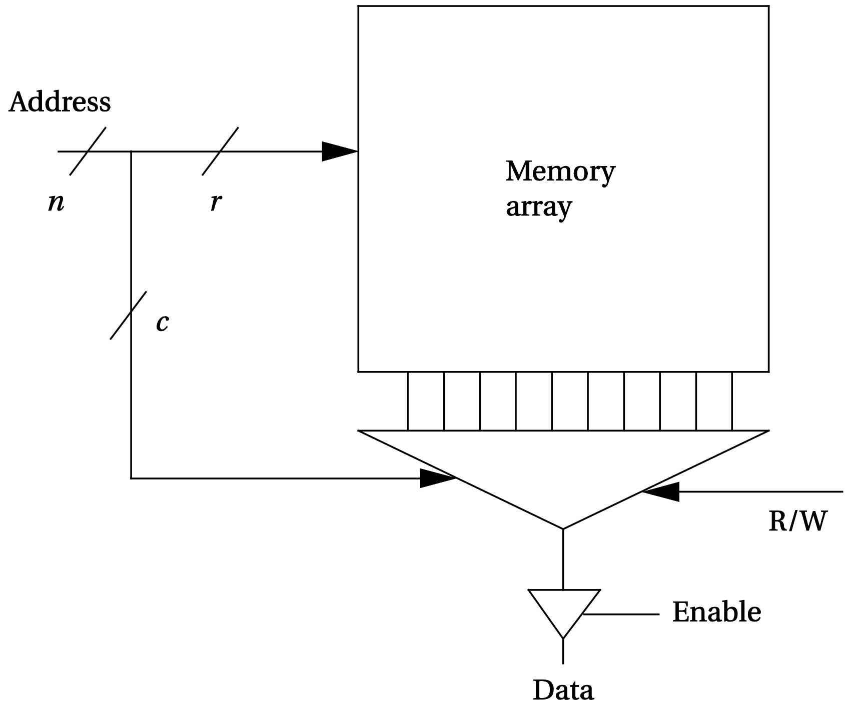


Key AMBA specifications

CHI Coherent Hub Interface	Credited coherent protocol Layered architecture for scalability
ACE AXI Coherency Extensions	ACE is a superset of AXI — system-wide coherency across multicore clusters
AXI Adv. eXtensible Interface	AXI supports separate A/D phases, bursts, multiple outstanding addresses, OoO responses
AHB Adv. High-performance Bus	AHB supports 64/128 bit multi-manager AHB-Lite for single manager
APB Adv. Peripheral Bus	System bus for low bandwidth peripherals

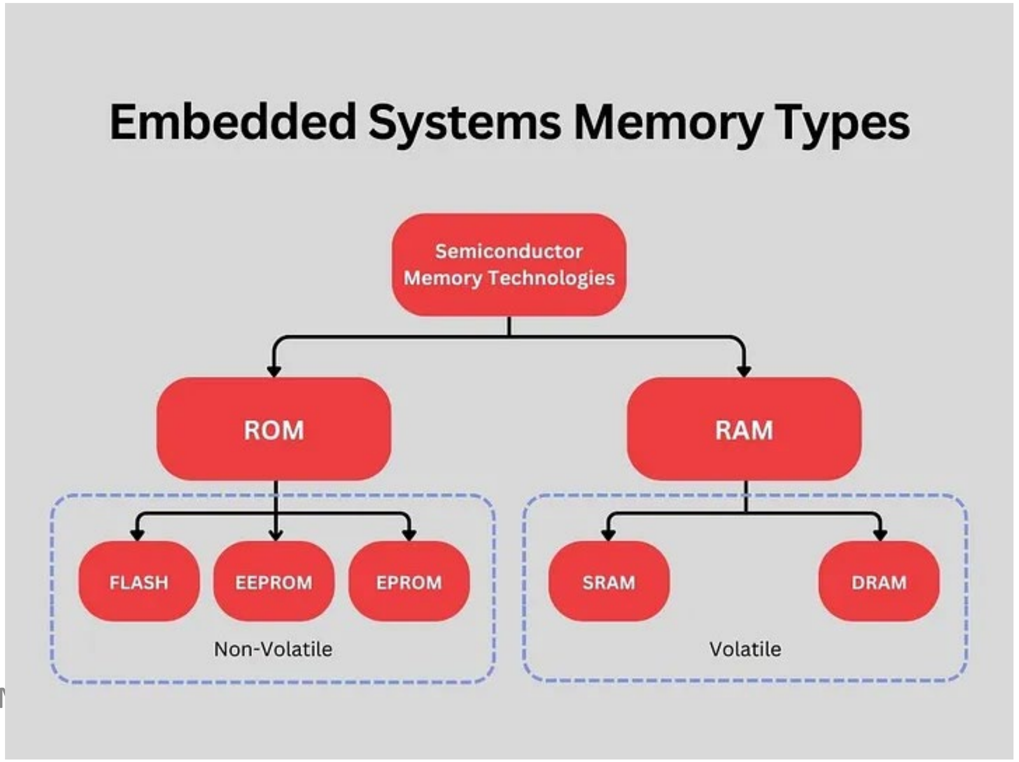
Memory components

- Several different types of memory:
  - DRAM.
  - SRAM.
  - Flash.
- Each type of memory comes in varying:
  - Capacities.
  - Widths.



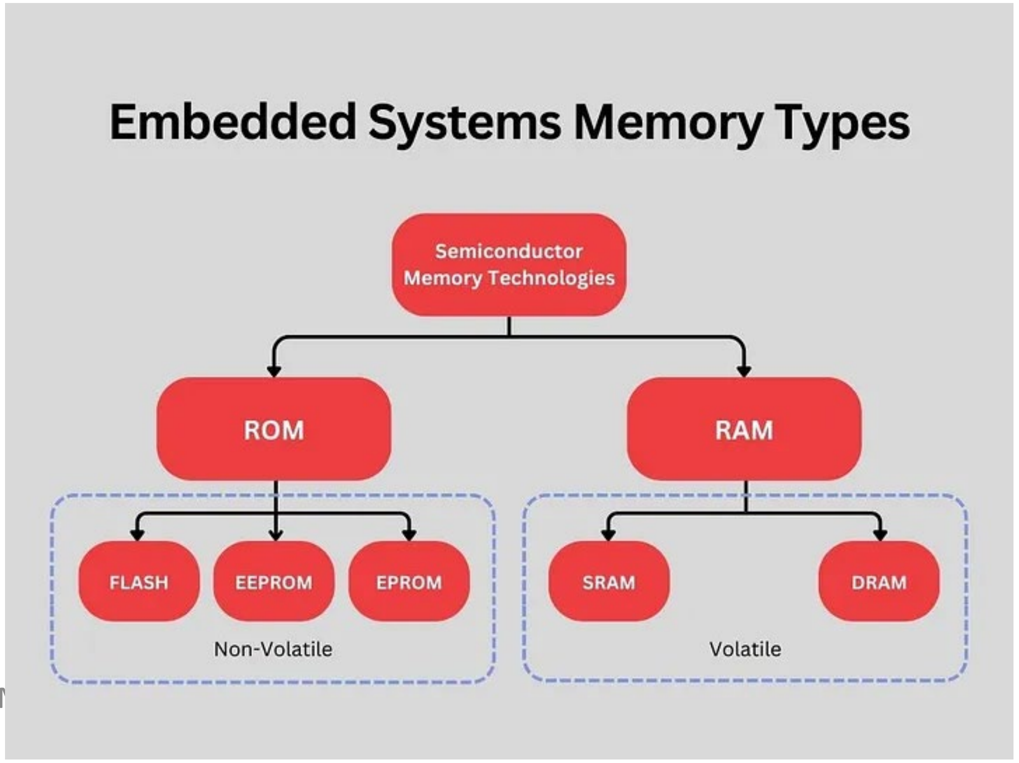
Memory

- Flash memory is a non-volatile, electrically reprogrammable storage solution. It offers high-density, low-cost storage with fast read times. Flash memory is widely used in embedded systems due to its numerous advantages, such as its ability to maintain data without power and quick access to stored data.
- Pros:
  - High-density storage
  - Low cost
  - Fast read times
  - Non-volatile, retaining data without power
  - Electrically reprogrammable
- Cons:
  - Erasing data is limited to one sector at a time
  - Slower write times compared to RAM
  - Finite number of write/erase cycles


Embedded Systems Memory Types
Semiconductor Memory Technologies
ROM
RAM
FLASH
EEPROM
EPROM
SRAM
DRAM
Non-Volatile
Volatile

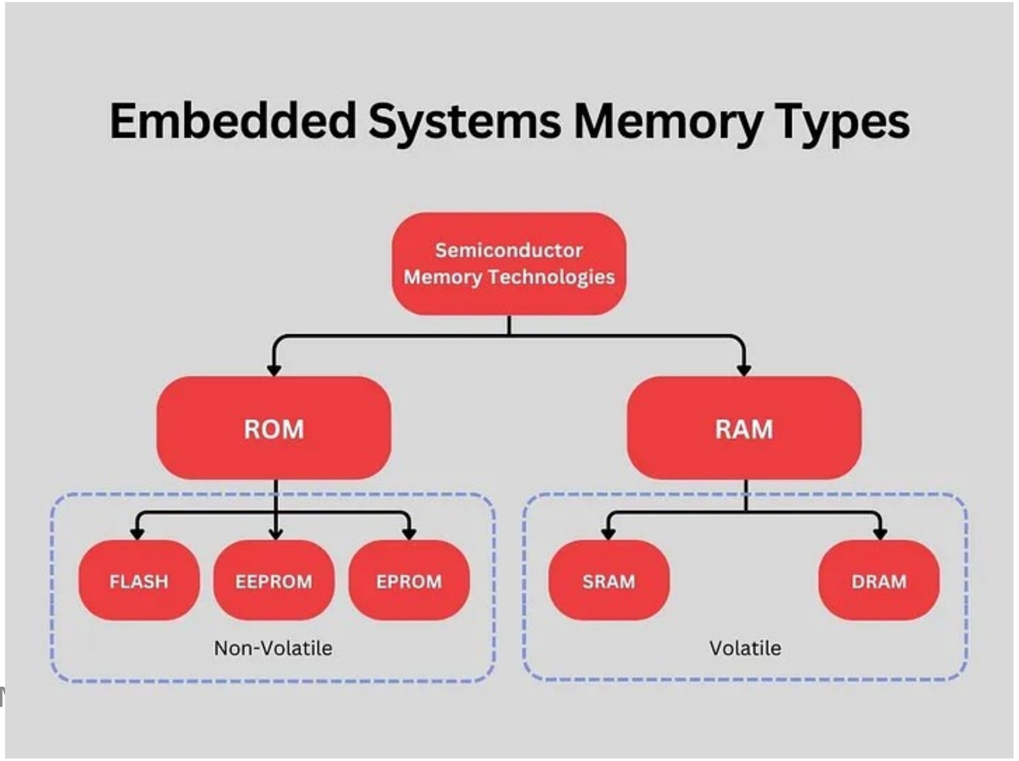
Memory

- Static RAM (SRAM) is a type of volatile memory that provides fast access times. It retains data as long as power is supplied to the system. SRAM is commonly used for high-speed caches in embedded systems.
- Pros:
    - Fast access times
    - No refresh cycles required, unlike DRAM
- Cons:
    - Higher cost-per-byte compared to DRAM
    - Consumes more power than DRAM
    - Requires more transistors per memory cell,
    - resulting in a larger chip size


Embedded Systems Memory Types
Semiconductor Memory Technologies
ROM
RAM
FLASH
EEPROM
EPROM
SRAM
DRAM
Non-Volatile
Volatile

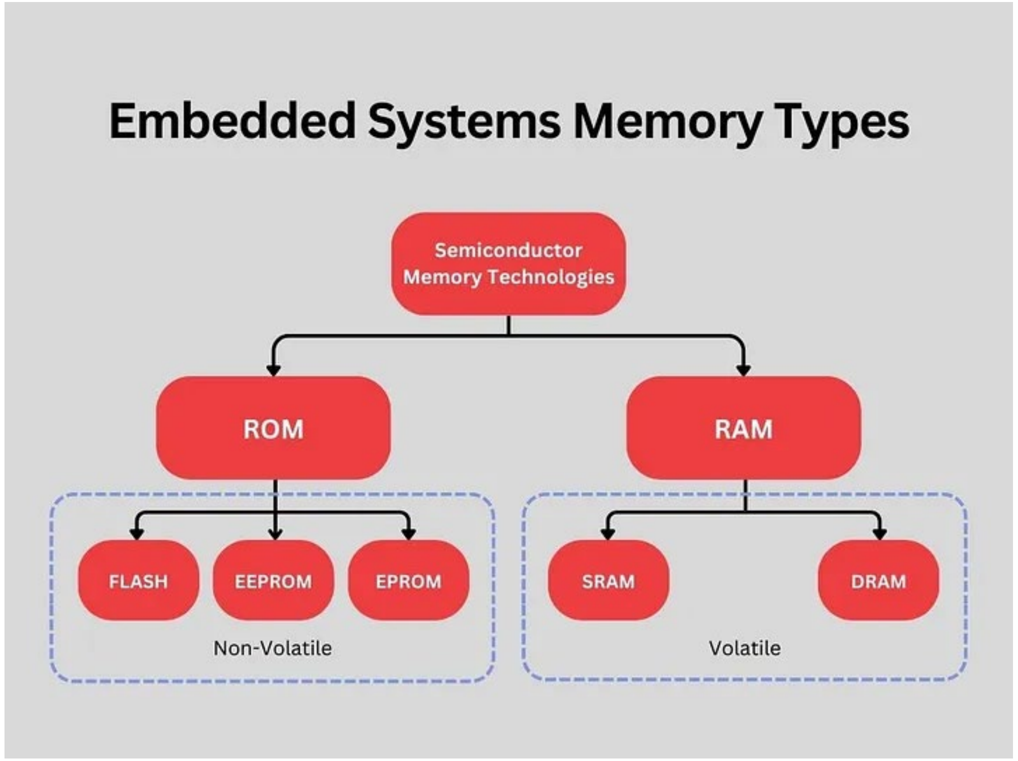
Memory

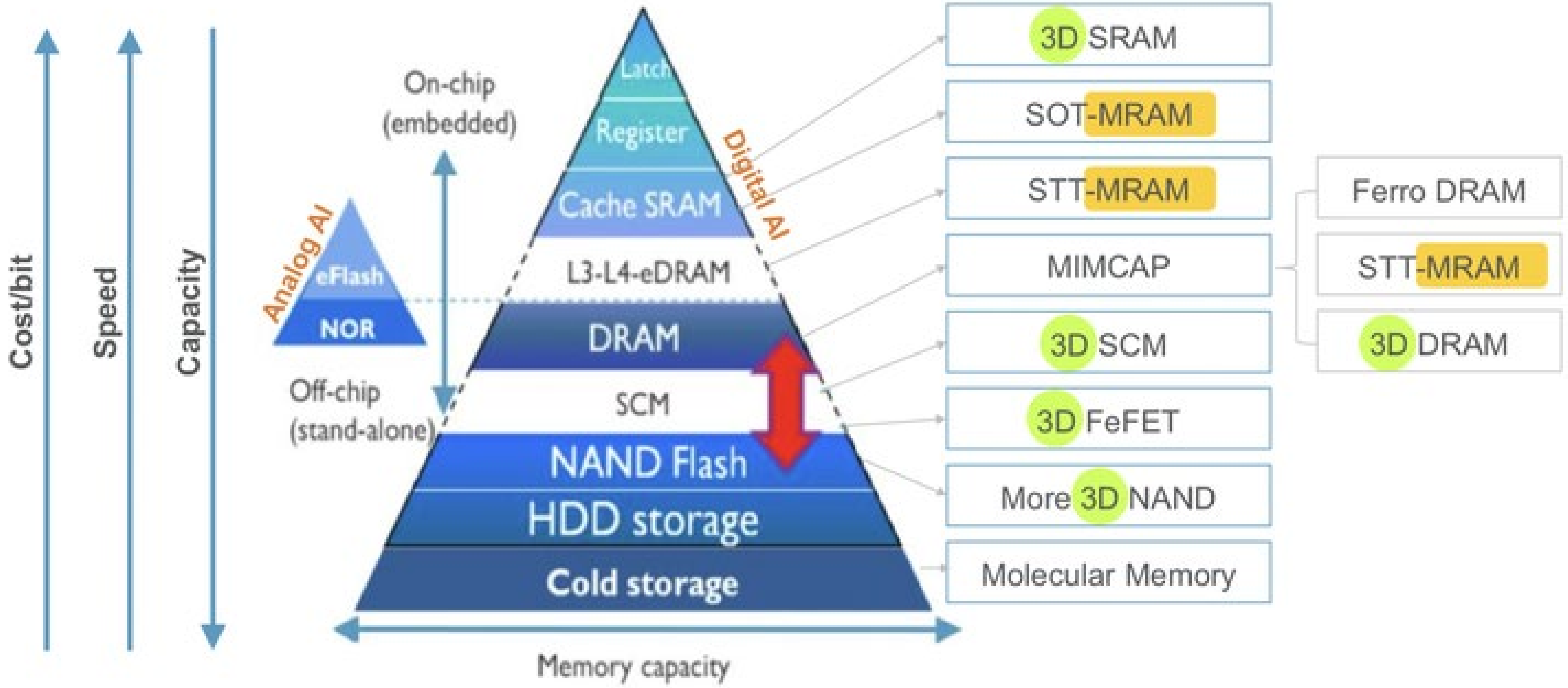
- Electrically-Erasable-Programmable Read-Only Memory (EEPROM) is a hybrid memory device that combines features of both RAM and ROM. It can be read and written like RAM, but it retains its contents without power, like ROM. EEPROMs can be erased and reprogrammed electrically, offering more flexibility than EPROMs.
- Pros:
  - Non-volatile, retaining data without power
  - Electrically erasable and reprogrammable
  - Allows byte-by-byte erasing and writing
- Cons:
  - Higher cost compared to other memory types
  - Longer write cycles than RAM
  - Limited number of write/erase cycles


Embedded Systems Memory Types
Semiconductor Memory Technologies
ROM
RAM
FLASH
EEPROM
EPROM
SRAM
DRAM
Non-Volatile
Volatile

Memory

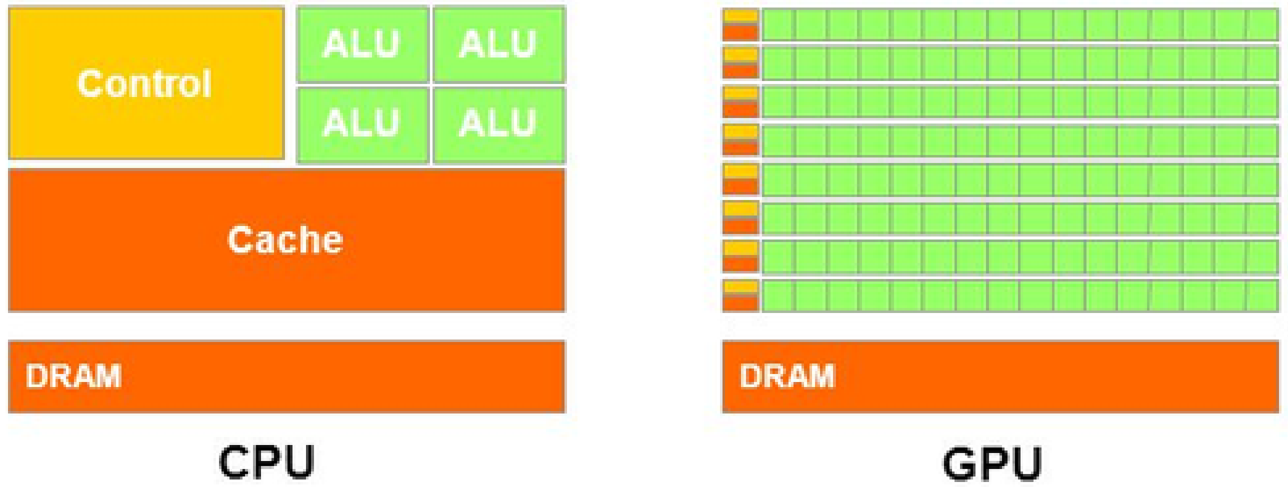
- Flash memory:
  - Widely used in embedded systems for firmware storage, data logging, and file systems due to its non-volatile nature, high-density storage, and low cost.
- SRAM:
  - Often employed as cache memory or for high-speed data paths in embedded systems, thanks to its fast access times. It is also suitable for low-power applications due to its simple internal structure.
- EEPROM:
  - Typically utilized for storing small amounts of non-volatile data, such as configuration settings, calibration data, and device IDs. Its byte-by-byte erasability and reprogrammability allow for easy updates and modifications.


Embedded Systems Memory Types
Semiconductor Memory Technologies
ROM
RAM
FLASH
EEPROM
EPROM
SRAM
DRAM
Non-Volatile
Volatile


Cost/bit
Speed
Capacity
On-chip (embedded)
Off-chip (stand-alone)
Analog AI
eFlash
NOR
Latch
Register
Cache SRAM
L3-L4-eDRAM
DRAM
SCM
NAND Flash
HDD storage
Cold storage
Digital AI
Memory capacity
3D SRAM
SOT-MRAM
STT-MRAM
MIMCAP
3D SCM
3D FeFET
More 3D NAND
Molecular Memory
Ferro DRAM
STT-MRAM
3D DRAM

CPU and GPU

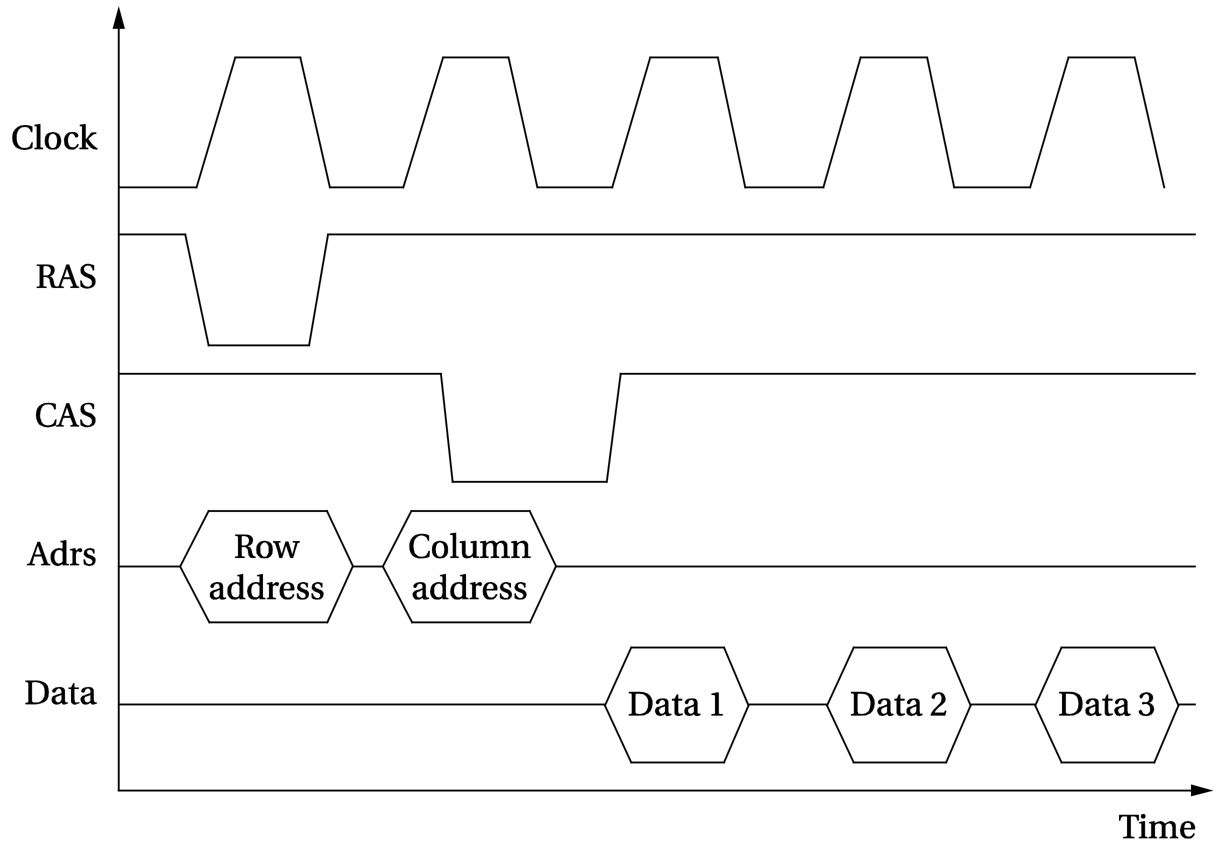
- GPU is specialized for compute intensive, highly data parallel computation
  - More area is dedicated to processing
  - Good for high arithmetic intensity programs with a high ratio between arithmetic operations and memory operations.



Random-access memory

- Dynamic RAM is dense, requires refresh.
    - SDRAM: synchronous DRAM.
    - EDO DRAM: extended data out.
    - FPM DRAM: fast page mode.
    - DDR DRAM: double-data rate.
- Static RAM is faster, less dense, consumes more power.

SDRAM read operation

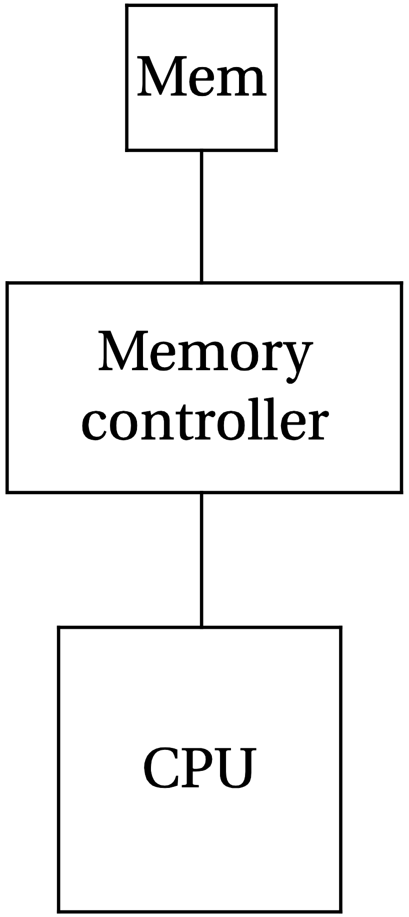


Memory packaging

- SIMM: single in-line memory module.
- DIMM: dual in-ilen memory module.

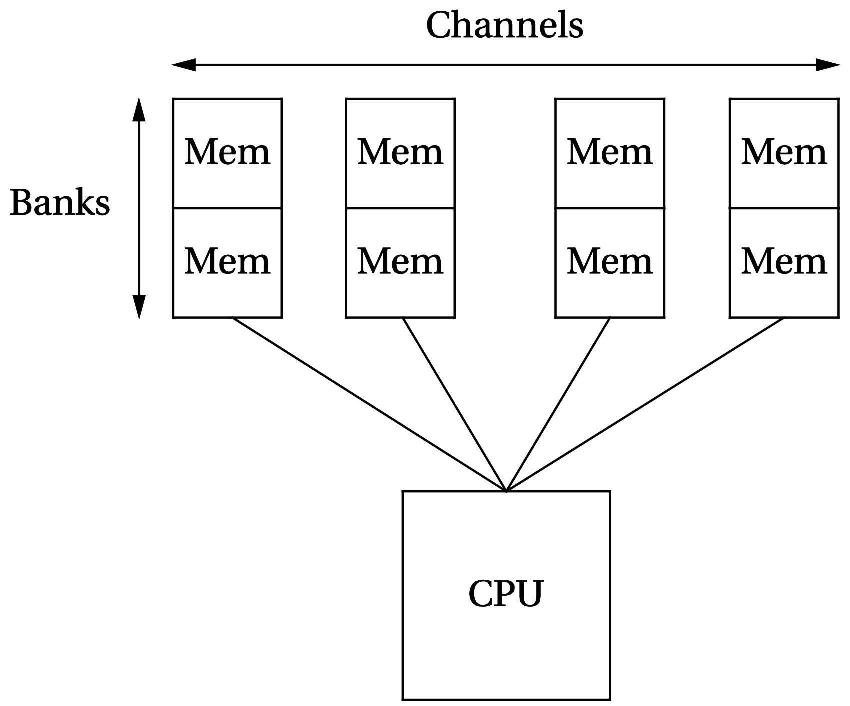
Memory systems and memory controllers

- Memory has complex internal organization.
- Memory controller hides details of memory interface, schedules transfers to maximize performance.



Channels and banks

- Channels provide separate connections to parts of memory.
- Banks are separate memory arrays.



Counters and timers

- Counter is a register with counting logic.
- Can be used to count events.
- May be up/down.
- Typically provides reset.

- Timer is a counter with a periodic input.
- A real-time clock provides time relative to seconds.
    - Hardware may count relative to a known start time.

GPIO

- General-purpose I/O (GPIO) provides simple input.
- Read value in input mode.
- Write value in output mode.
  - Value will be held at pin.
- May provide several different logic levels, drive capabilities, etc.

Data conversion

- Convert between analog and digital.
- Characteristics:
    - Rate at which conversion can be made.
    - Precision in bits.
    - Accuracy relative to a standard.
- Analog/digital conversion (ADC) offers a wide range of cost/performance/precision trade-offs.
- Digital/analog conversion (DAC) is more straightforward but money buys quality.

Examples

- Automotive radar ADC operates at 1.6 gigasamples per second (GSPS) and 12 bits of resolution.
- Instrumentation ADC converts 32 bits at up to 38.4 KSPS.
- Radio frequency DAC converts 14 bits at 9 GSPS, 14 bits at 3 GSPS over a range 10 MHz-6 GHz.
- Instrumentation DAC converts 20- bits of resolution.

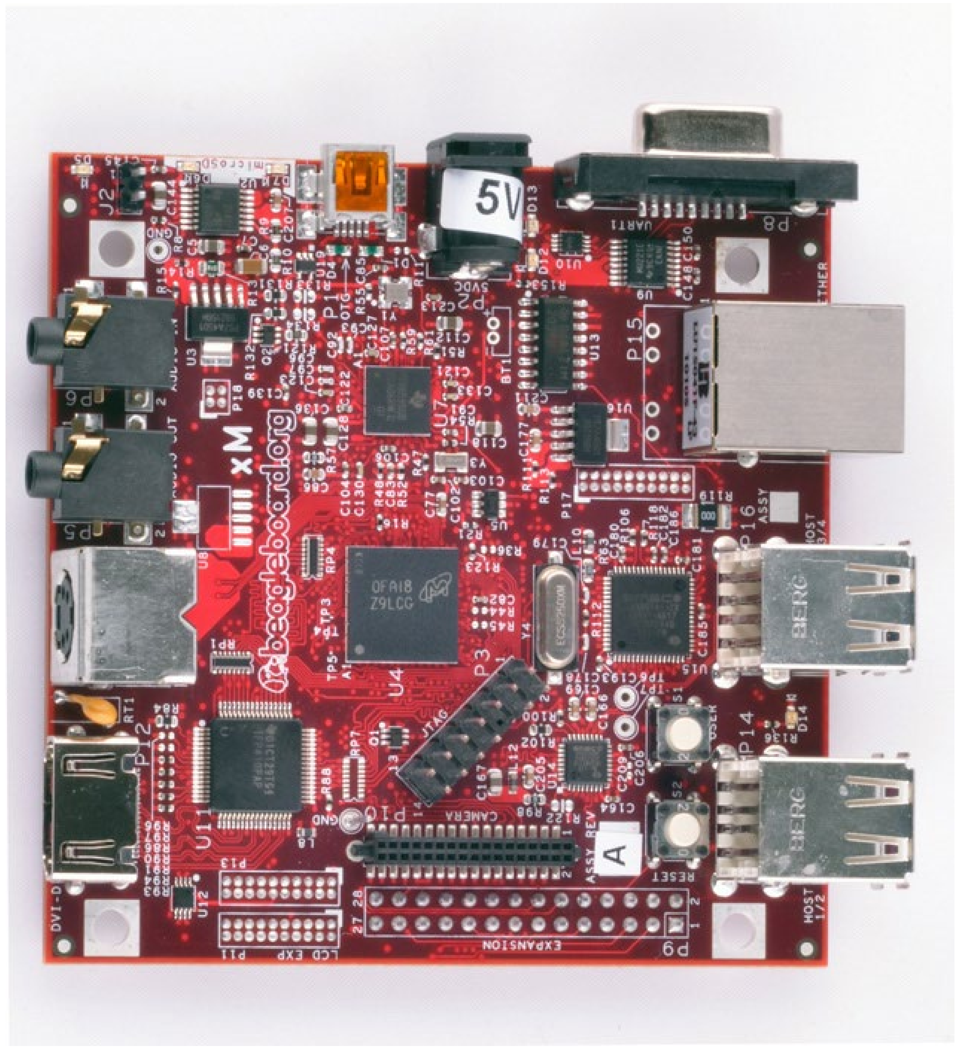
Computing platforms

- Design methodology.
- Consumer electronics architectures.
- System-level performance and power analysis.
- Platform security.

Evaluation boards

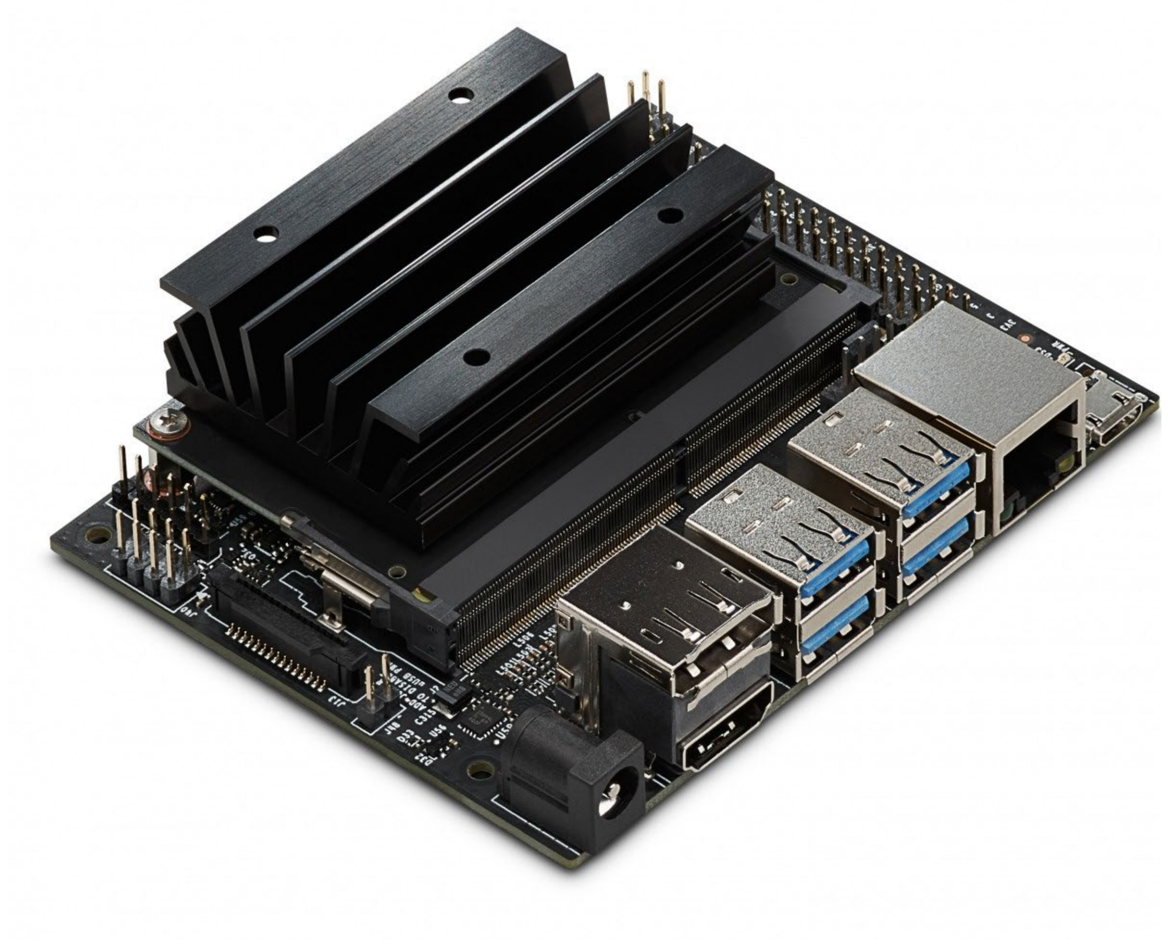
- Designed by CPU manufacturer or others.
- Includes CPU, memory, some I/O devices.
- May include prototyping section.
- CPU manufacturer often gives out reference design---can be used as starting point for your custom board design.

BeagleBoard



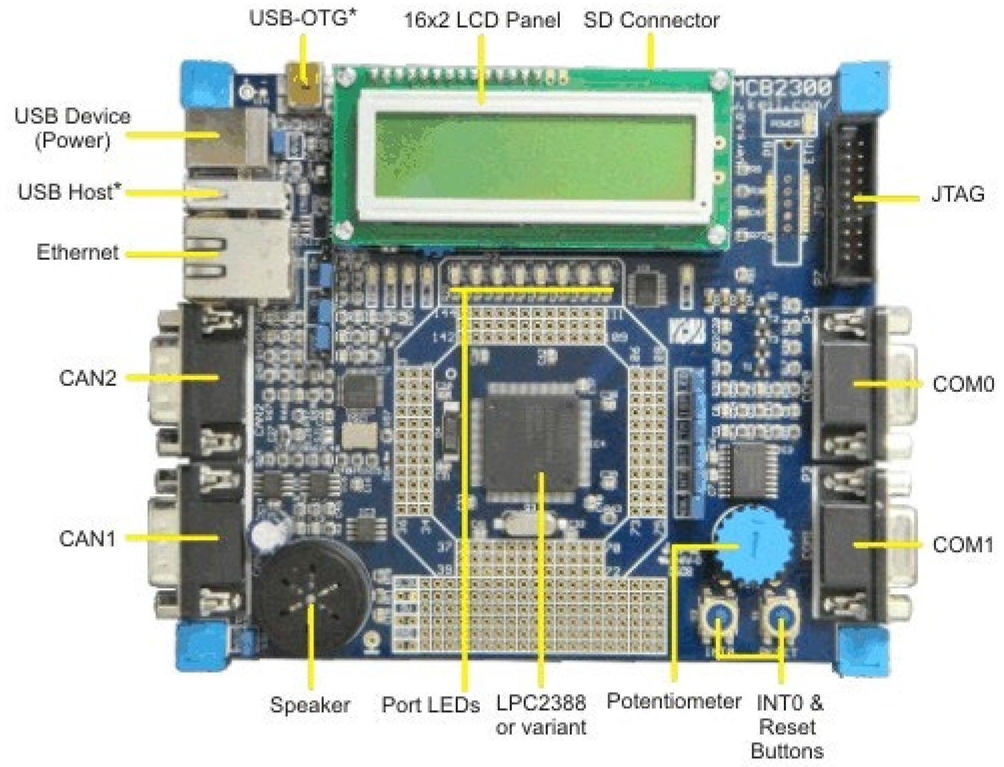
- OMAP processor.
- Audio input and output.
- Video output.
- SD card.

NVIDIA Jetson Nano



- a quadcore Arm core
- a 128-core NVIDIA Maxwell GPU
- video encoder and decoder
- I/O includes four USB 3.0 ports, a MIPI camera port, an HDMI port, and Gigabit Ethernet
- The system boots off a microSD card

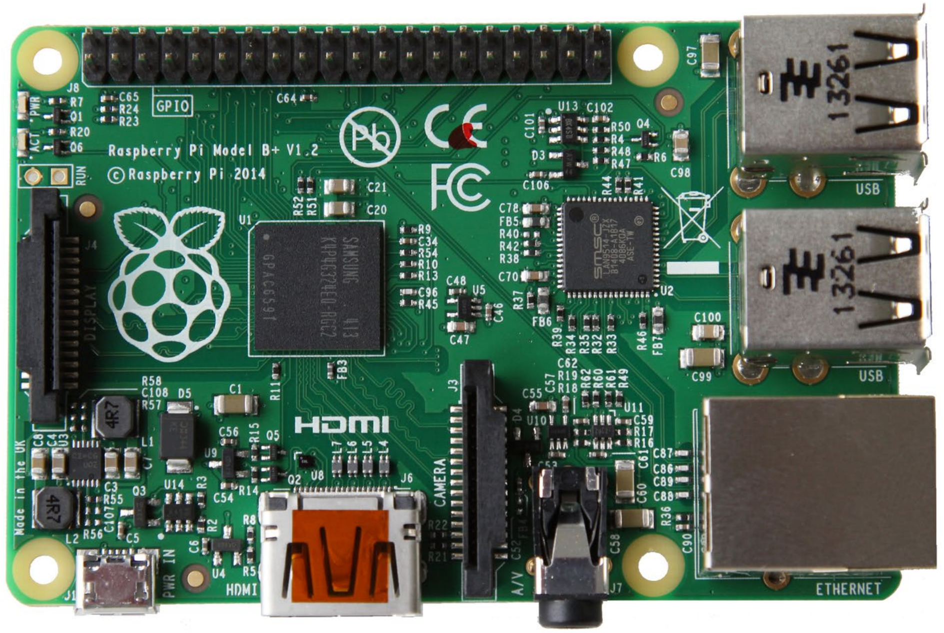
ARM evaluation module

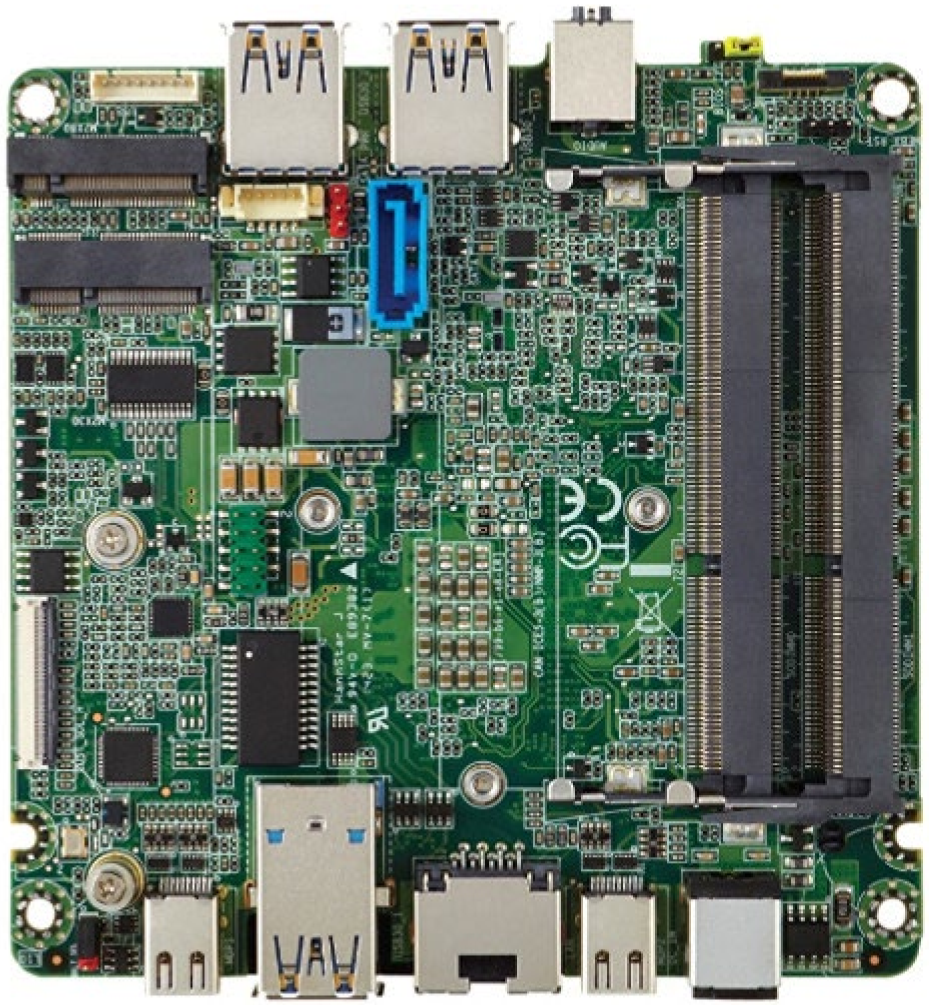


* MCB2388 and later MCB2387 boards only

- ARM processor.
- Display, serial port, etc.
- Prototyping area.

Others


Raspberry Pi Model B+ V1.2
©Raspberry Pi 2014
GPIO
HDMI
CAMERA
DISPLAY
USB
ETHERNET
A/V
PWR IN
Made in the UK



Choosing a platform

- CPU: choice of instruction sets, features, etc.
- Bus determines available I/O devices, system performance.
- Memory size, speed.
- I/O devices vary in performance, cost.

Intellectual property

- Hardware designs, source or object code, netlists, etc.
- Used at all levels of design:
  - Schematics for hardware reference design.
  - Drivers and run-time libraries.
  - Software development environments.

BeagleBoard IP

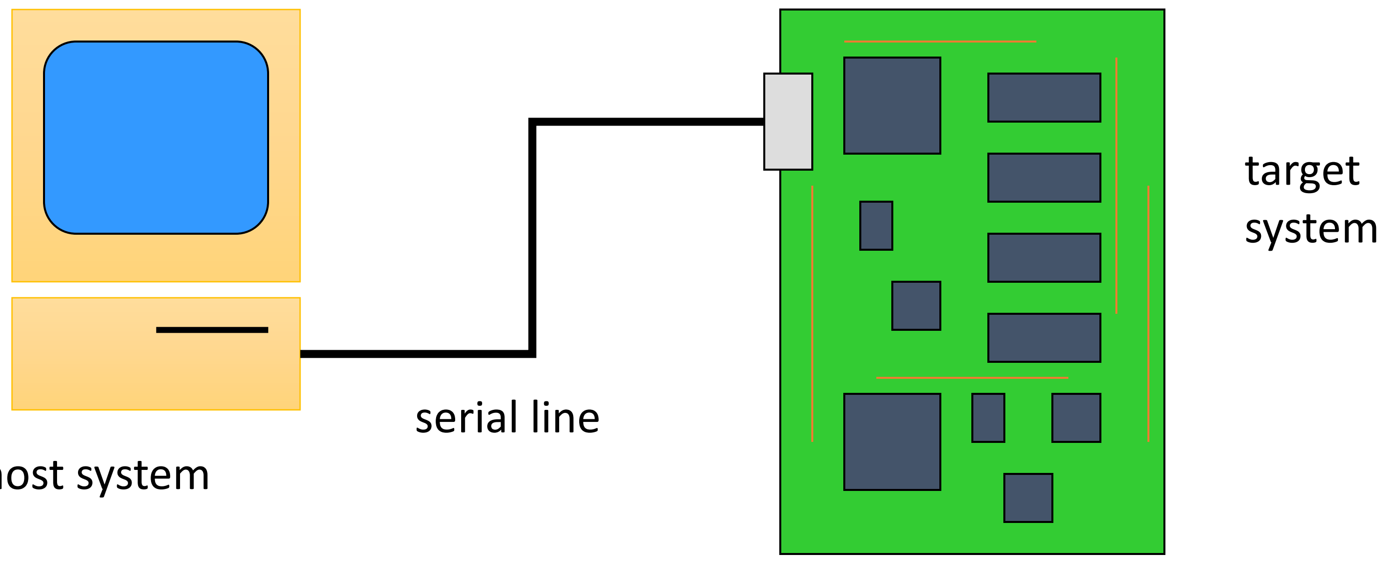
- PCB schematics and artwork files.
- Bill of materials for components.
- Compiler.
- Linux.

Debugging embedded systems

- Challenges:
  - target system may be hard to observe;
  - target may be hard to control;
  - may be hard to generate realistic inputs;
  - setup sequence may be complex.

Host/target design

- Use a host system to prepare software for target system:



Host-based tools

- Cross compiler:
  - compiles code on host for target system.
- Cross debugger:
  - displays target state, allows target system to be controlled.

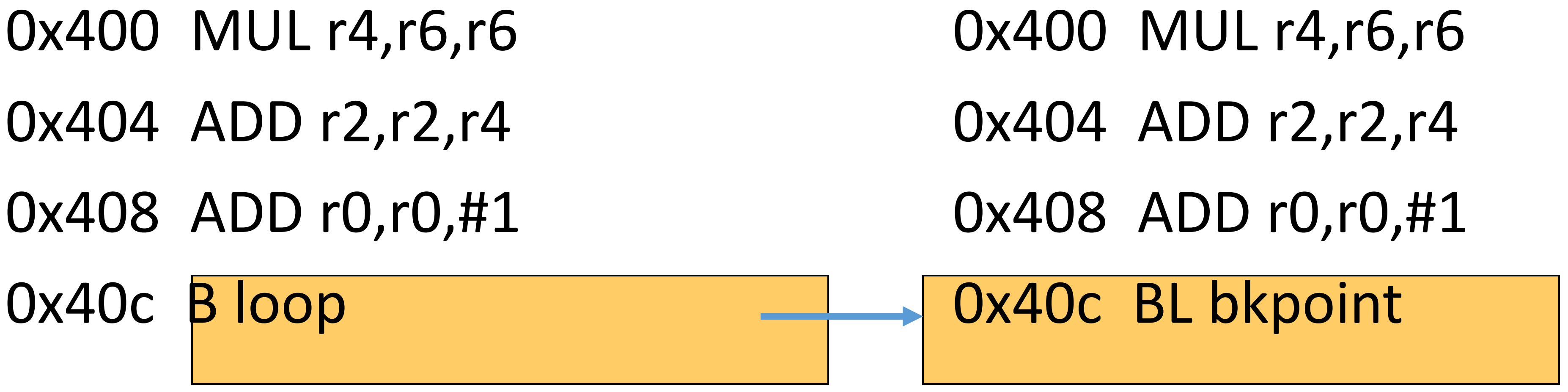
Software debuggers

- A monitor program residing on the target provides basic debugger functions.
- Debugger should have a minimal footprint in memory.
- User program must be careful not to destroy debugger program, but , should be able to recover from some damage caused by user code.

Breakpoints

- A breakpoint allows the user to stop execution, examine system state, and change state.
- Replace the breakpointed instruction with a subroutine call to the monitor program.

ARM breakpoints



```
0x400  MUL r4,r6,r6
0x404  ADD r2,r2,r4
0x408  ADD r0,r0,#1
0x40c  B loop
```

uninstrumented code

```
0x400  MUL r4,r6,r6
0x404  ADD r2,r2,r4
0x408  ADD r0,r0,#1
0x40c  BL bkpoint
```

code with breakpoint

Breakpoint handler actions

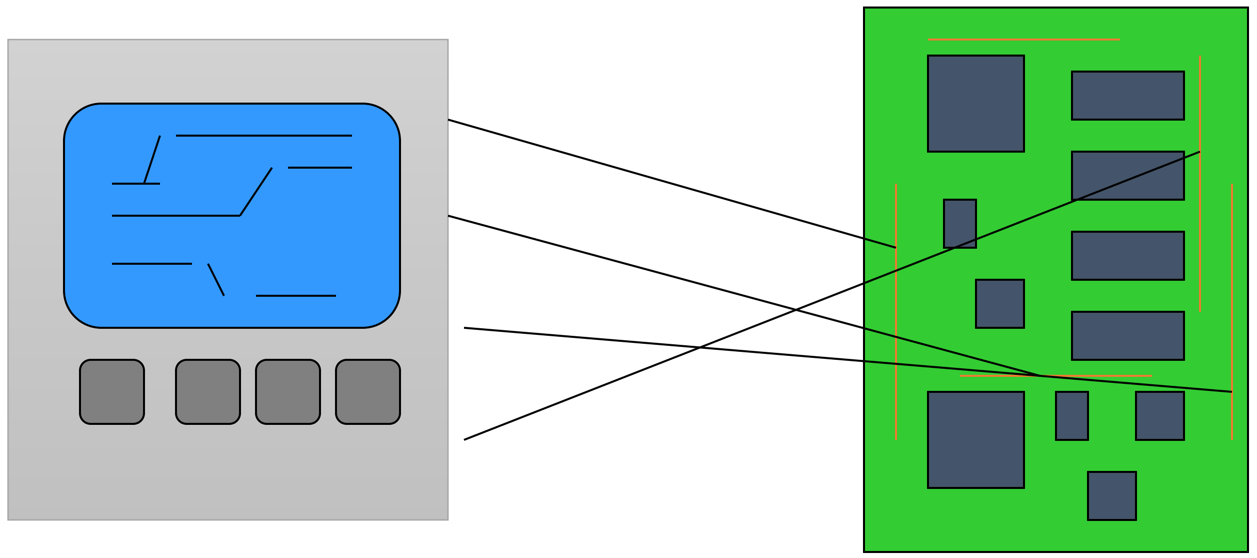
- Save registers.
- Allow user to examine machine.
- Before returning, restore system state.
  - Safest way to execute the instruction is to replace it and execute in place.
  - Put another breakpoint after the replaced breakpoint to allow restoring the original breakpoint.

In-circuit emulators

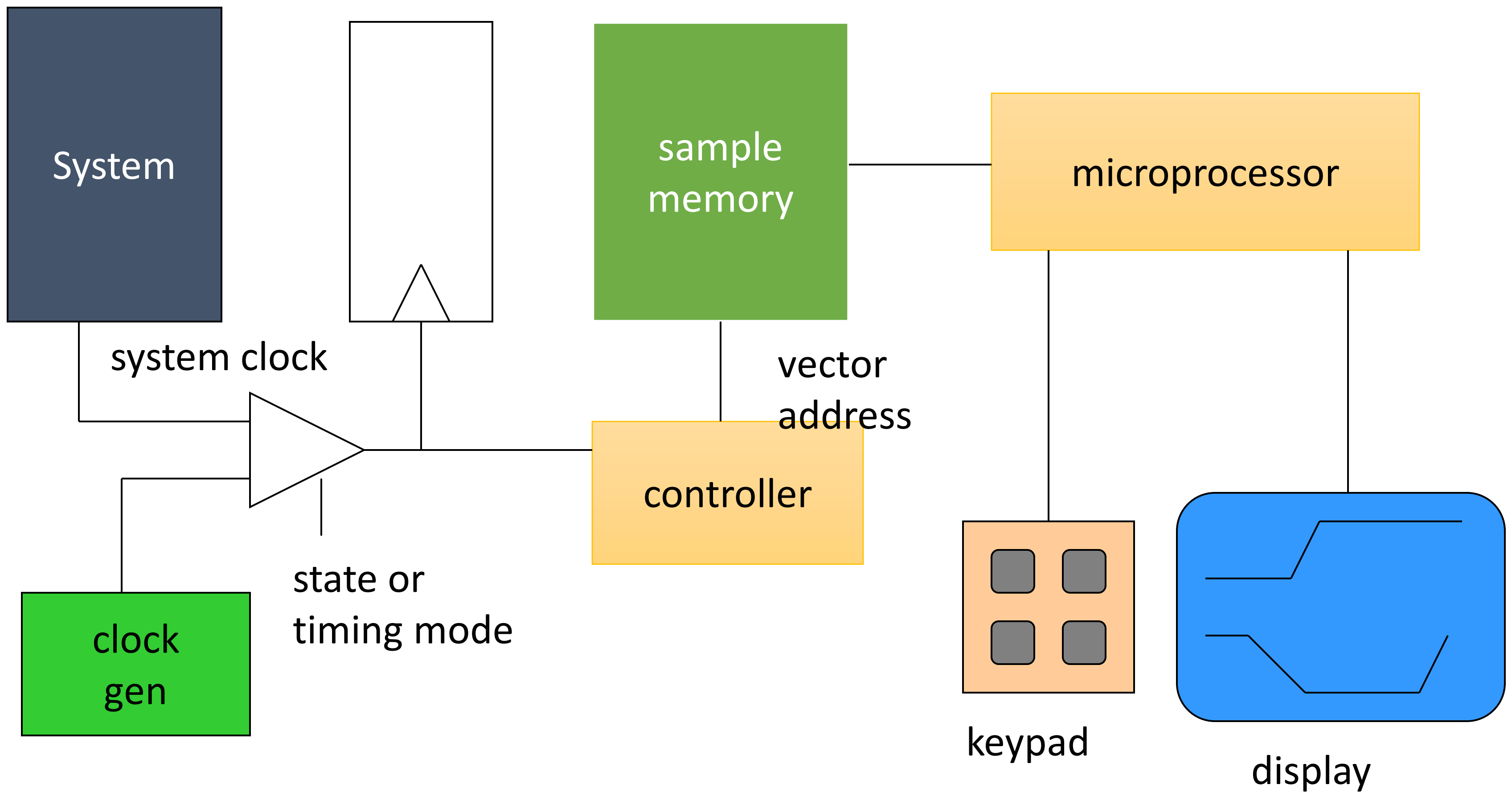
- A microprocessor in-circuit emulator is a specially-instrumented microprocessor.
- Allows you to stop execution, examine CPU state, modify registers.

Logic analyzers

- A logic analyzer is an array of low-grade oscilloscopes:

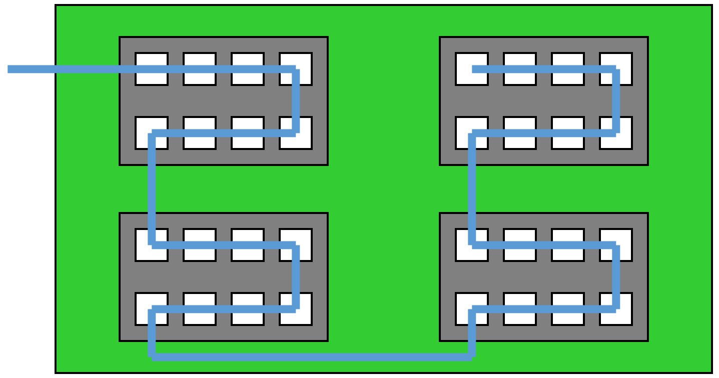


Logic analyzer architecture



Boundary scan

- Simplifies testing of multiple chips on a board.
  - Registers on pins can be configured as a scan chain.
  - Used for debuggers, in-circuit emulators.

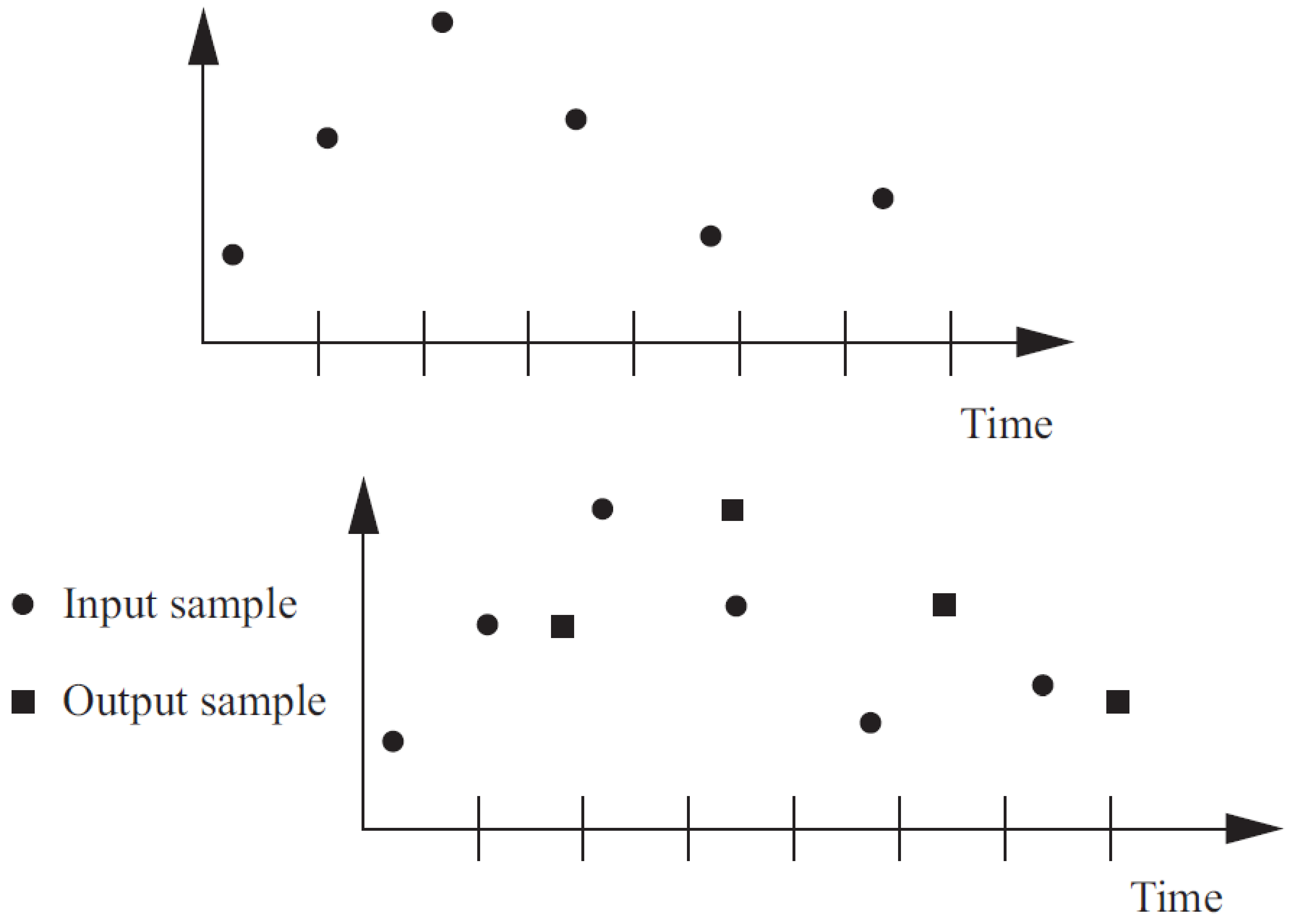


How to exercise code

- Run on host system.
- Run on target system.
- Run in instruction-level simulator.
- Run on cycle-accurate simulator.
- Run in hardware/software co-simulation environment.

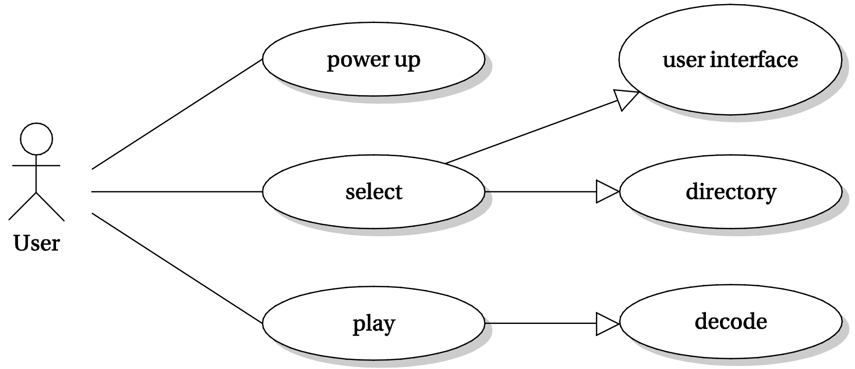
Debugging real-time code

- Bugs in drivers can cause non-deterministic behavior in the foreground problem.
- Bugs may be timing-dependent.


Time
Input sample
Output sample
Time

Consumer electronics use cases

- Multimedia: stored in compressed form, uncompressed on viewing.
- Data storage and management: keep track of your multimedia, etc.
- Communication: download, upload, chat.

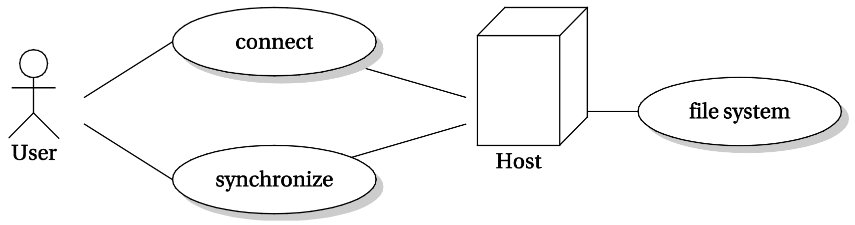


Non-functional requirements for CE

- Often battery-operated, strict power budget.,
- Very inexpensive.
- User interface must be capable but inexpensive.

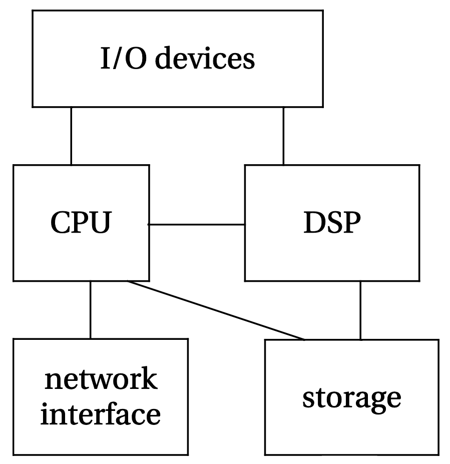
CE devices and hosts

- Many devices talk to host system.
  - PC host does things that are hard to do on the device.
- Increasingly, CE devices communicate directly over the network, avoiding the host for access.



Platforms and operating systems

- Many CE devices use a DSP for signal processing and a RISC CPU for other tasks.
- I/O devices include buttons, screen, USB.

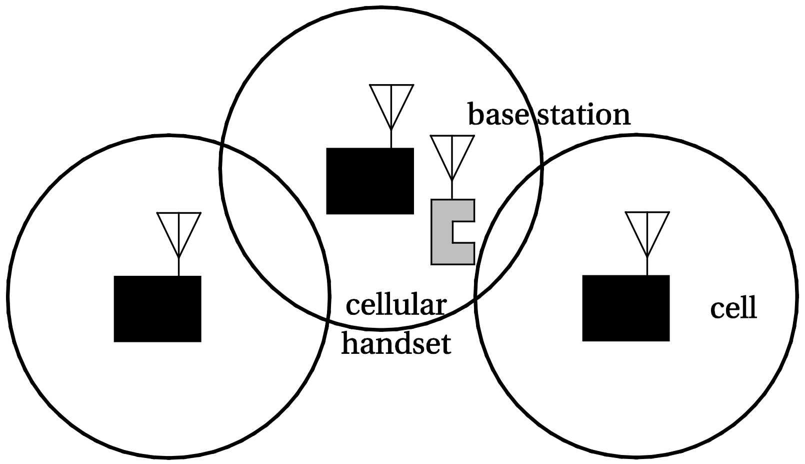


Flash file systems

- Flash is widely used for mass storage.
- Flash wears out on writing (up to 1 million cycles).
    - Directory is most often written, wears out first.
- Flash file system has layer that moves contents to levelize wear.
    - Hides wear leveling from API.

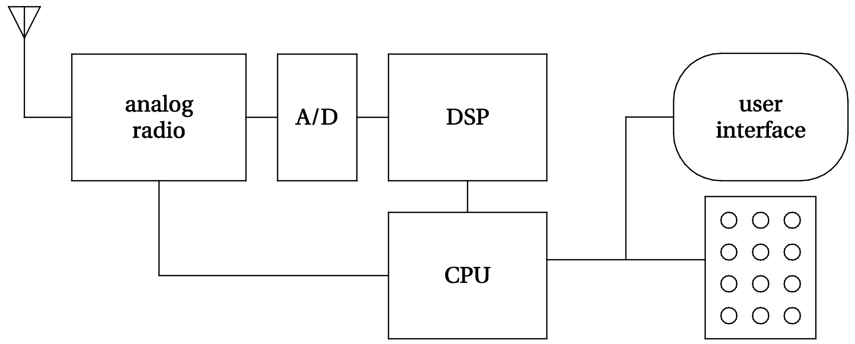
Cell phones

- Most popular CE device in history; most widely used computing device.
  - 1 billion sold per year.
- Handset talks to cell.
- Cells hand off handset as it moves.



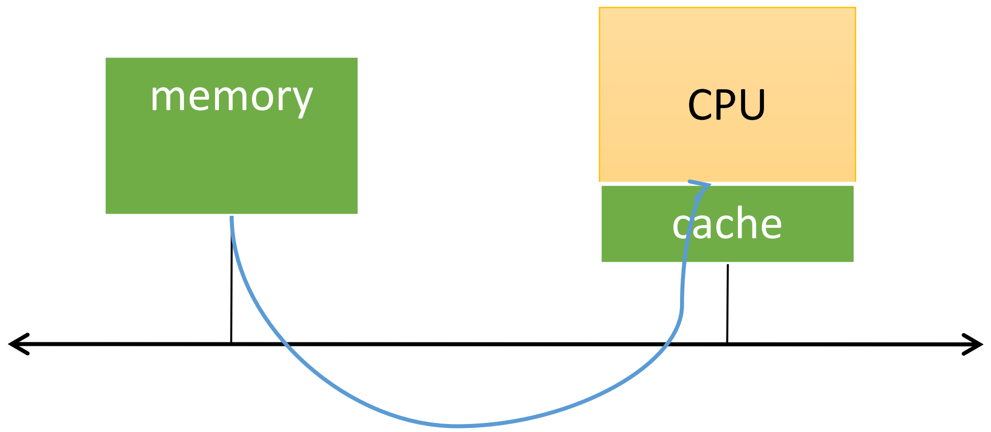
Cell phone platforms

- Today’s cell phones use analog front end, digital baseband processing.
  - Future cell phones will perform IF processing with DSP.
- Baseband processing in DSP:
  - Voice compression.
  - Network protocol.
- Other processing:
  - Multimedia functions.
  - User interface.
  - File system.
  - Applications (contacts, etc.)



System-level performance analysis

- Performance depends on all the elements of the system:
  - CPU.
  - Cache.
  - Bus.
  - Main memory.
  - I/O device.



Bandwidth as performance

- Bandwidth applies to several components:
  - Memory.
  - Bus.
  - CPU fetches.
- Different parts of the system run at different clock rates.
- Different components may have different widths (bus, memory).

Bandwidth and data transfers

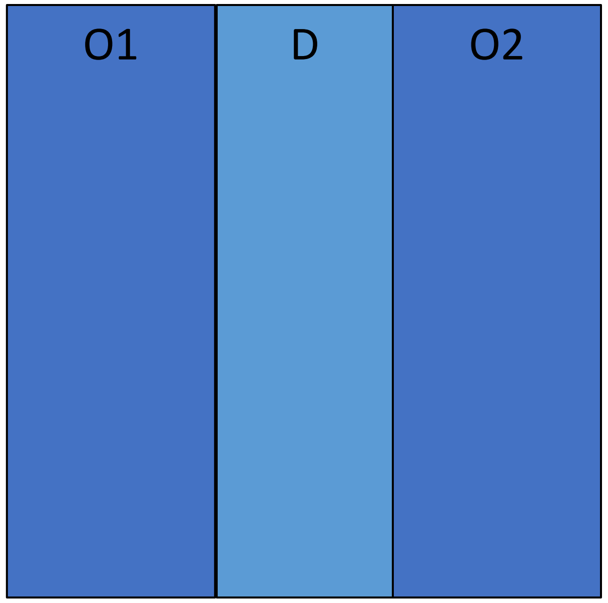
- Video frame: 1920 x 1080 x 3 = 6.2 MBytes.
  - Transfer in 1/30 sec = 0.033 sec,
- 100 MHz 8-bit wide bus requires 0.062 sec/frame---too slow.
- Increase bus width to 32 bits, transfer time reduced to 0.015 sec/frame.
- Increase bus speed to 200 MHz, transfer time is 0.031 frame/sec for8-bit wide bus.

Bus bandwidth modeling

- T: # bus cycles.
- P: bus cycle period.
- Total time for transfer:
  - t = TP.
- D: data transfer

Bus bandwidth

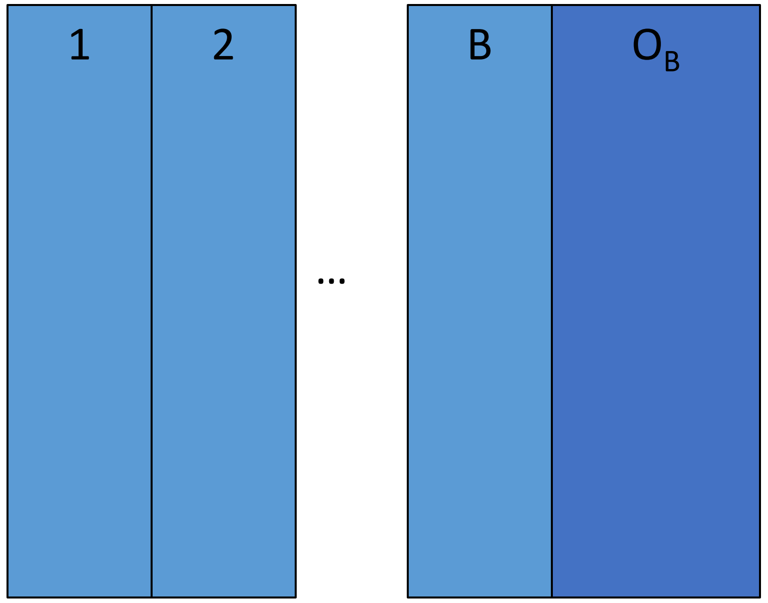
- Assume one word per transfer, compute time for N word transfer.
- Data transfer time D (may include wait states).
- Overhead O1 + O2 = O.



$$T_{basic}(N) = (O + D)N$$

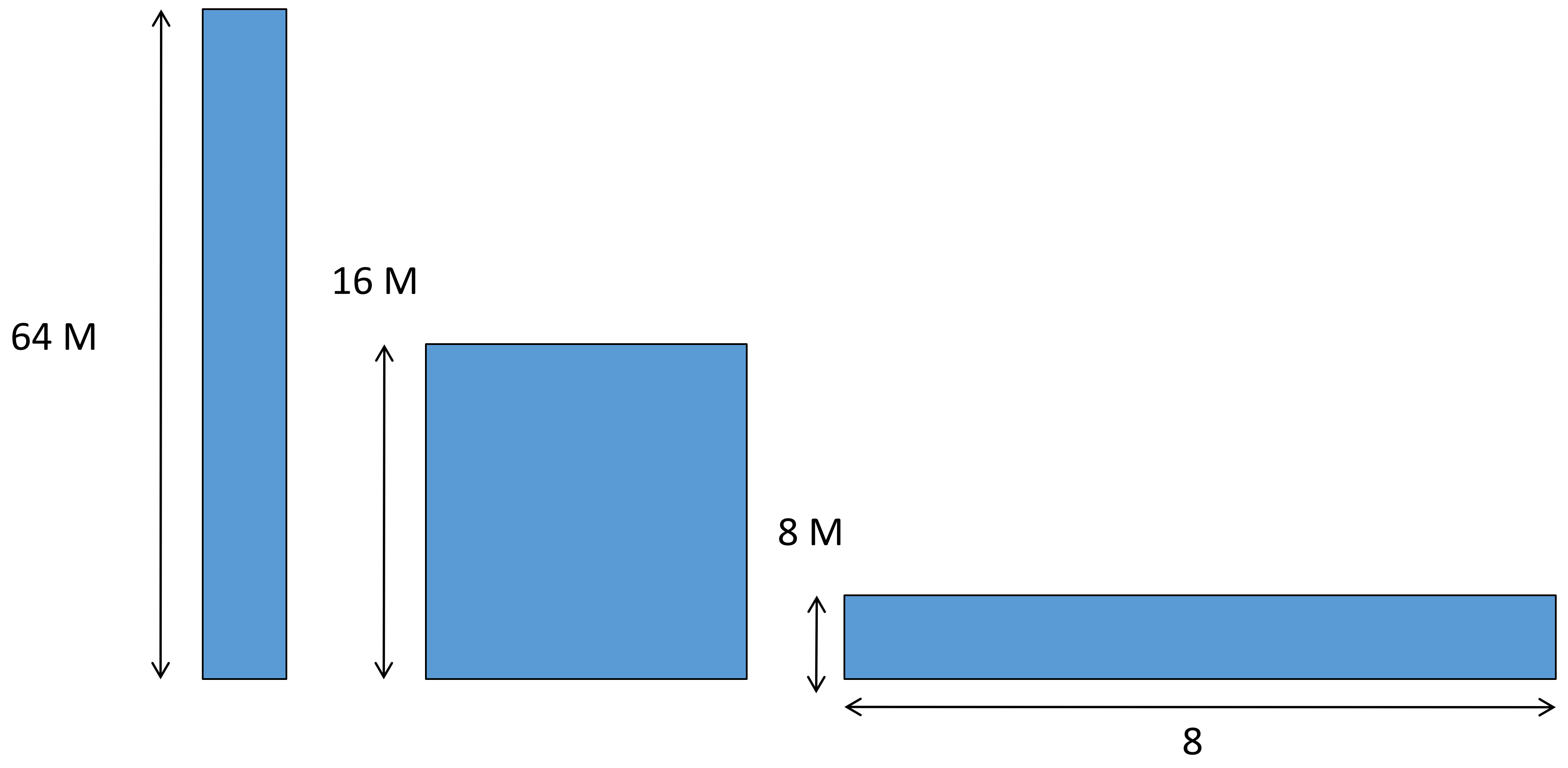
Bus burst transfer bandwidth

- B: transfers per burst.
  - N words.
- D clock cycles per transfer.
- O cycles overhead per burst.



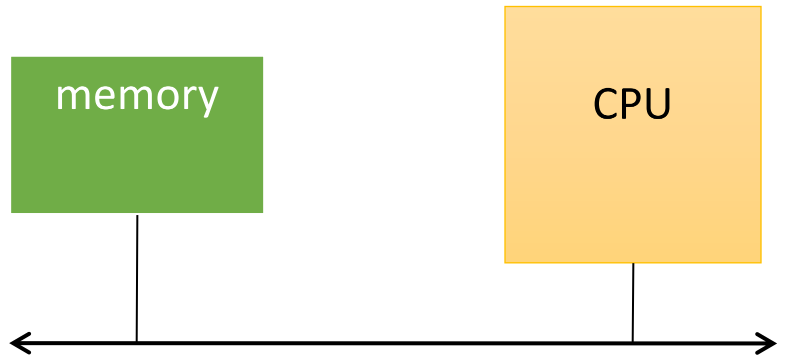
$$T_{burst}(N) = \frac{N}{B}(BD + O_B)$$

Memory aspect ratios



Bus performance bottlenecks

- Transfer 1920 x 1080 video frame @ 30 frames/sec = 6.2 MBytes/sec.
- Is performance bottleneck bus or memory?



Bus performance bottlenecks, cont'd.

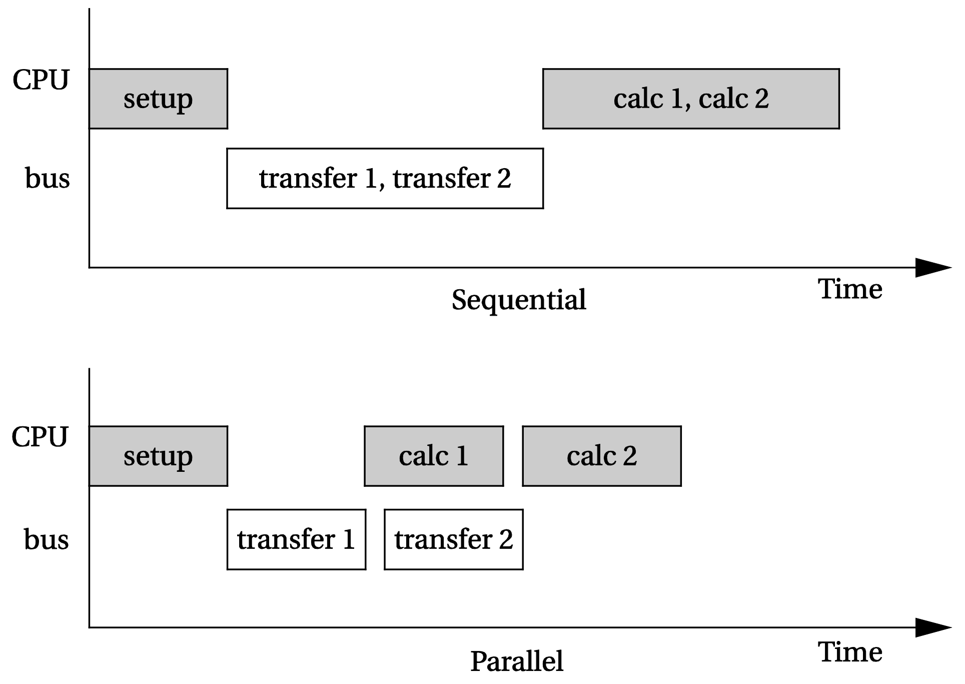
- Bus: assume 100 MHz bus, D=1, O=3:
    - $T_{basic}(1920 \times 1080) = (3 + 1) \cdot \left(\frac{6.2 \times 10^6}{2}\right) = 12.4 \times 10^6\ cycles$
    - $t_{basic} = T_{basic}P = 0.124\ sec$
- Memory: try burst mode B=4, two bytes wide.
    - $T_{basic}(1920 \times 1080) = \frac{(6.2 \times 10^6/2)}{4}(4 \cdot 1 + 4) = 6.2 \times 10^6\ cycles$
    - $t_{basic} = T_{basic}P = 0.062\ sec$

Bus performance spreadsheet

P	1.00E-07		
N	3110400		
O	3		
D	1		
B	1		
T	12441600		
t	1.24E+00		

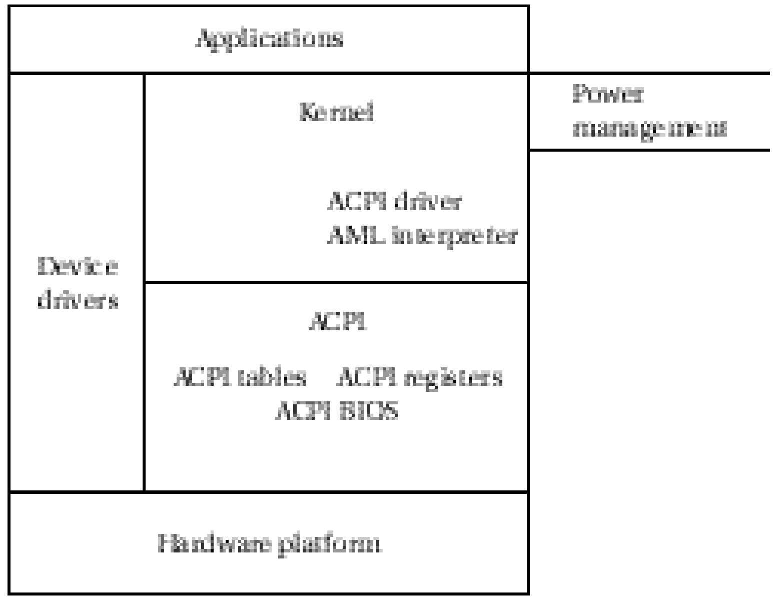
Parallelism

- Speed things up by running several units at once.
- DMA provides parallelism if CPU doesn't need the bus:
  - DMA + bus.
  - CPU.



ACPI

- Standard power management mechanism for PC.
  - OS determines actions, ACPI provides services.
- States:
  - G3, mechanical off.
  - G2, soft off.
  - G1, sleep.
  - G0, working.



Cryptography

- Cryptography is the science of hiding information.
- Secret-key cryptography allows messages to be encoded and decoded.
  - Key must be kept secret or the message will not be secure.
- Public-key cryptography gives secret messages with an easier-to-use approach:
  - Sender uses secret key to encrypt message. Also provides a public key to others.
  - Receiver can decrypt message using public key. The sender’s identity is established by the ability to use their public key.

Cryptographic hash and digital signature

- Cryptographic hash function generates a message digest.
    - Short version of the message.
    - Two different messages are unlikely to generate the same key.
- Digital signature:
    - Sender signs the message or message digest using private key.
    - Receiver decrypts with public key.
- Digital signature plus encryption:
    - Sender signs the message or message digest with private key.
    - Sender encrypts the message with the receiver’s public key.
    - Receiver decrypts with private key, then verifies signature using sender’s public key.

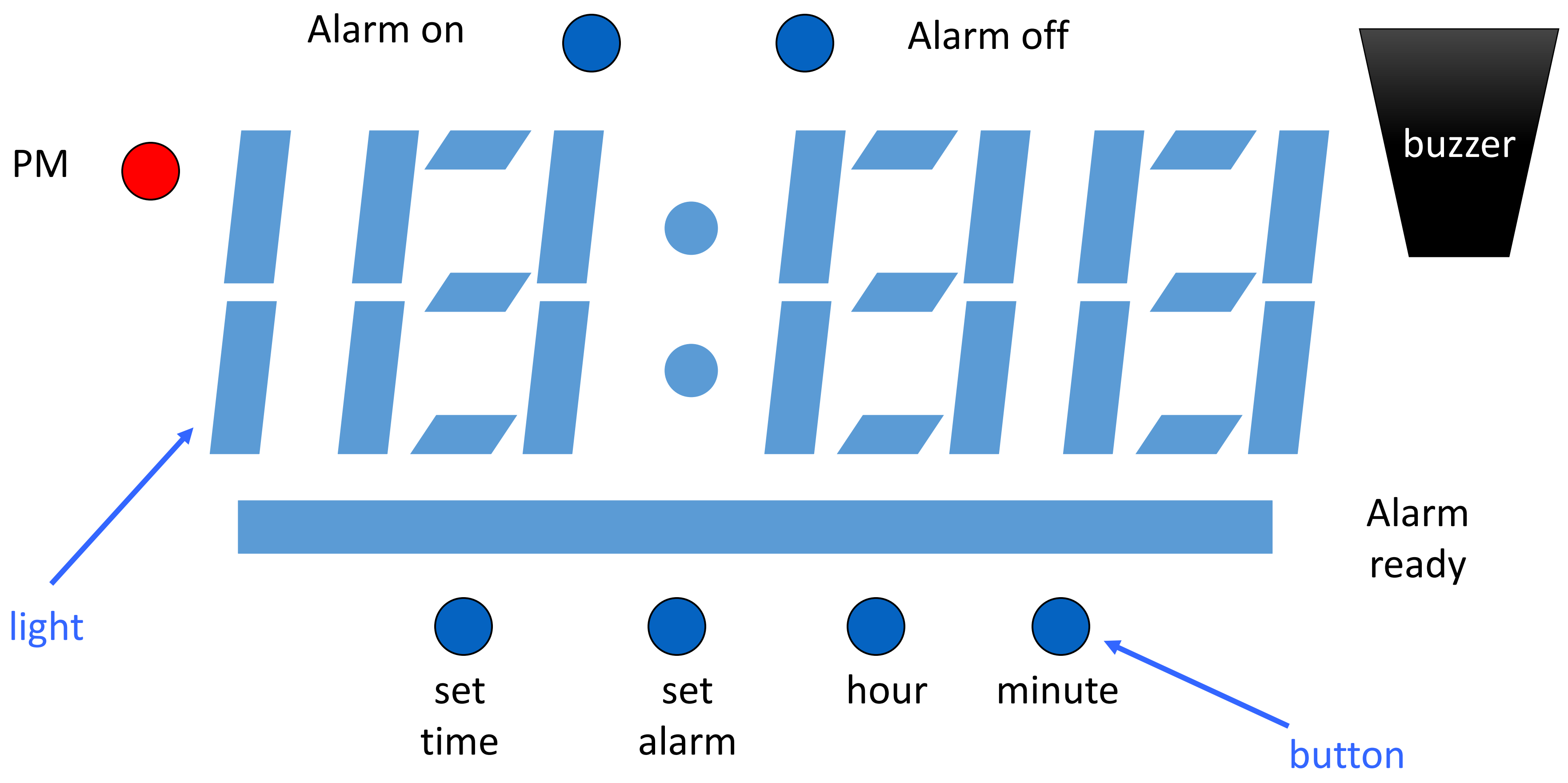
Attestation

- Allows Bill to check whether Alice has been compromised before sharing secure information with Alice.
- For example, hash certain code or data that will be known to Bill.
  - If that important information was changed, the hash sent by Alice will not match what Bill expects.

Computing Platforms

- Example: alarm clock

Alarm clock interface



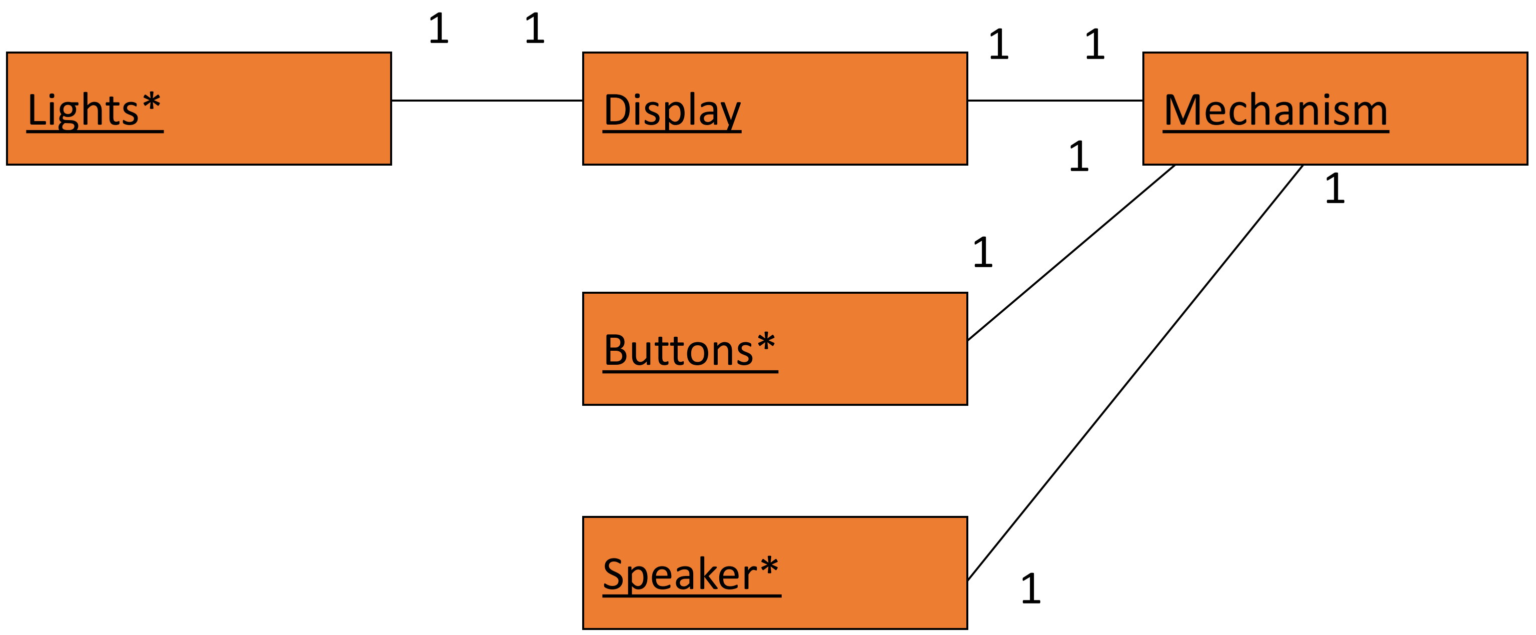
Operations

- Set time: hold set time, depress hour, minute.
- Set alarm time: hold set alarm, depress hour, minute.
- Turn alarm on/off: depress alarm on/off.

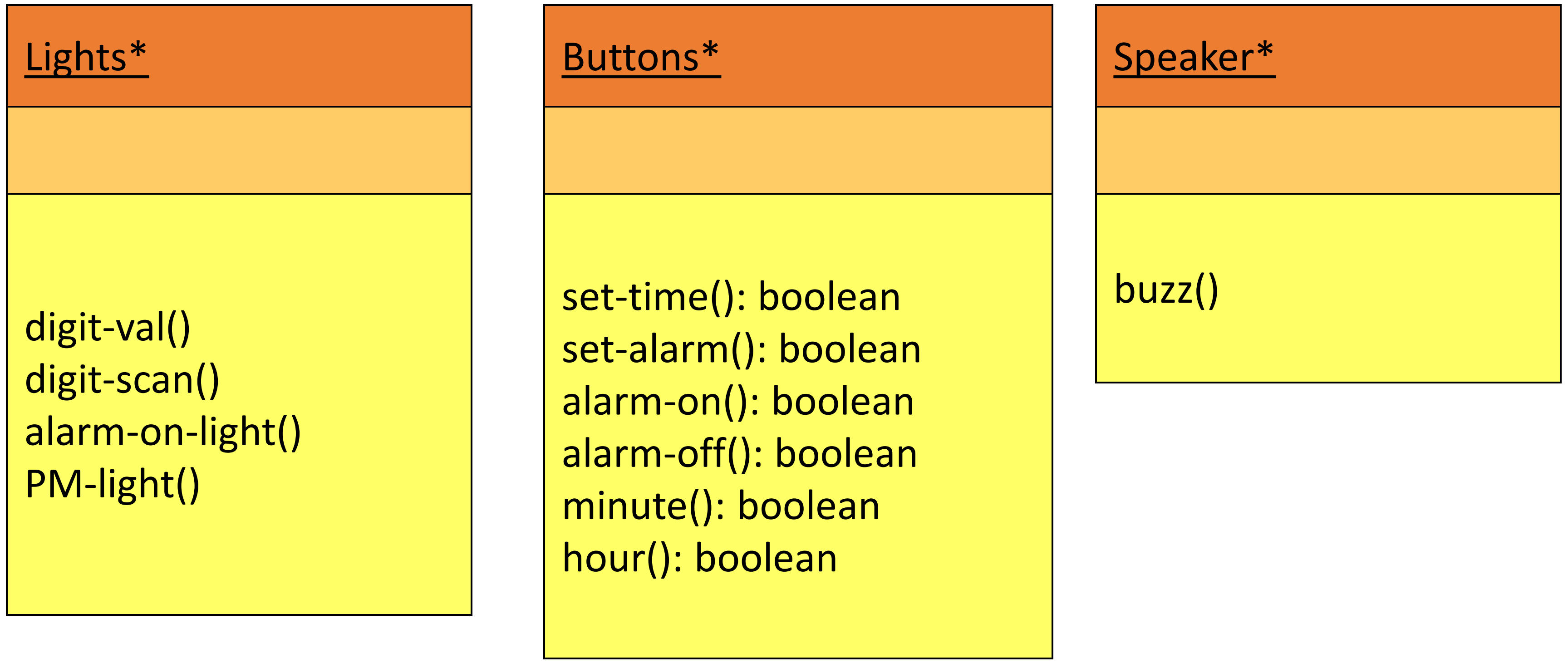
Alarm clock requirements

name	alarm clock
purpose	24-hour digital clock with one alarm
inputs	set time, set alarm, hour, minute, alarm on/off
outputs	four-digit display, PM indicator, alarm ready, buzzer
functions	keep time, set time, set alarm, turn alarm on/off, activate buzzer by alarm
performance	hours and digits, no seconds; not high precision
manufacturing cost	consumer product
power	AC
physical size/weight	fits on stand

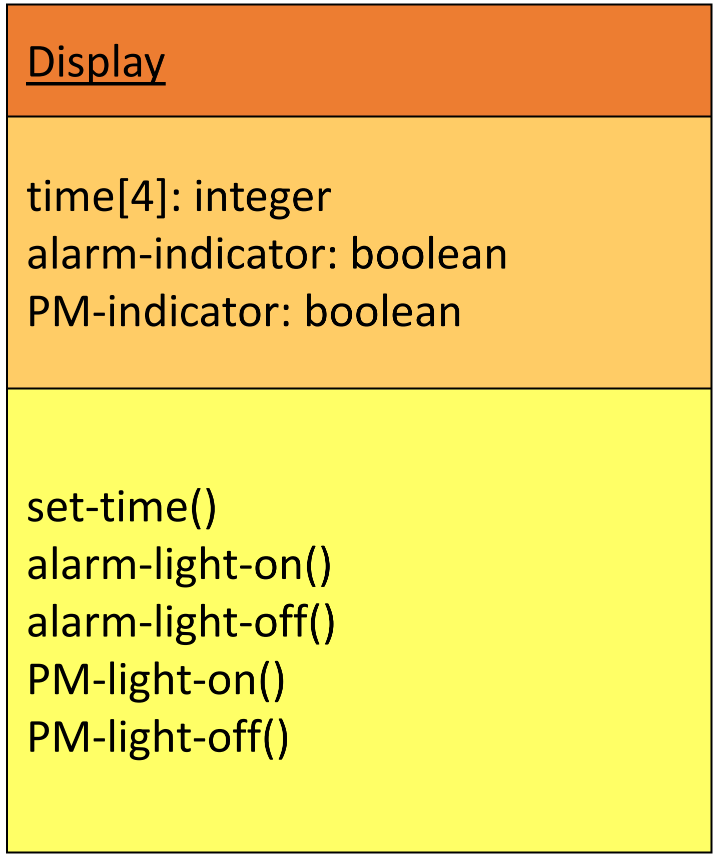
Alarm clock class diagram



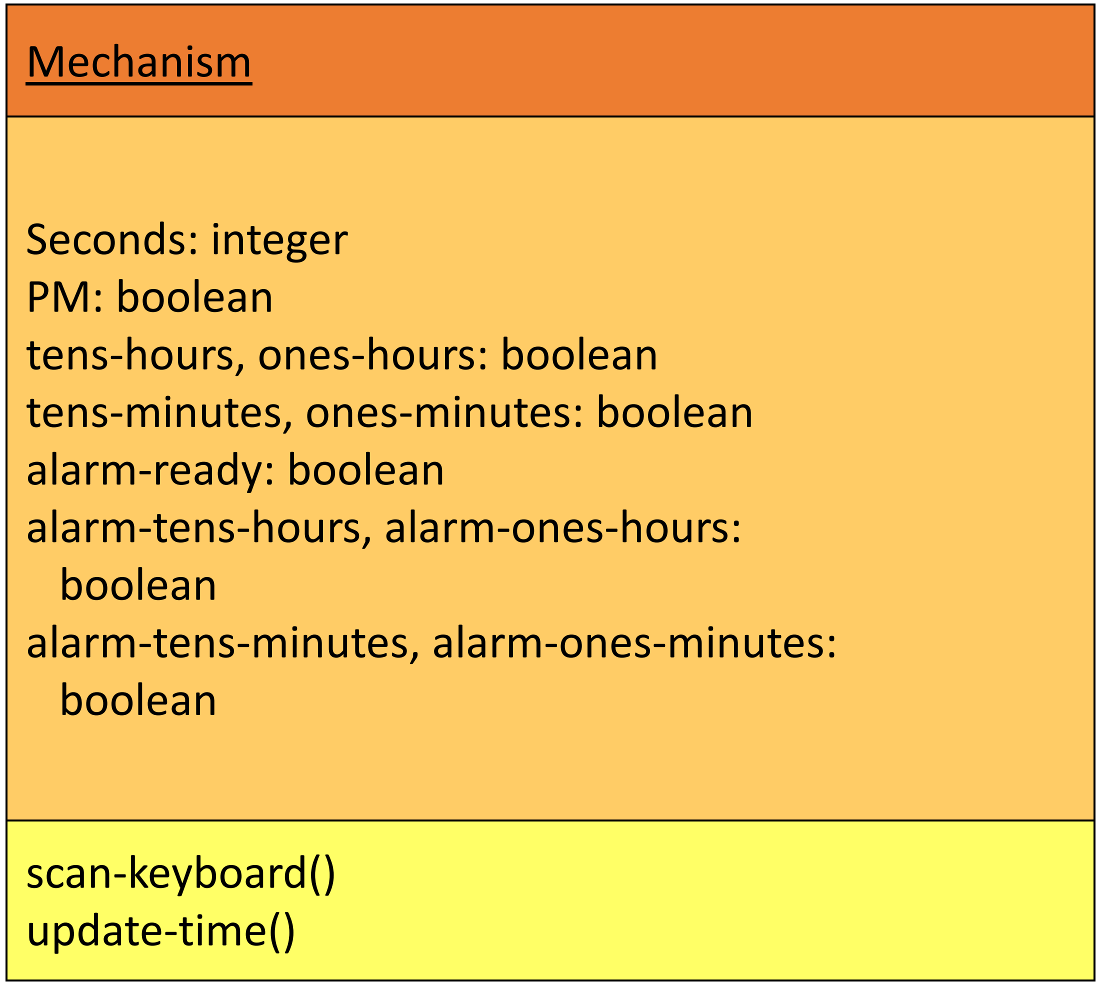
Alarm clock physical classes



Display class



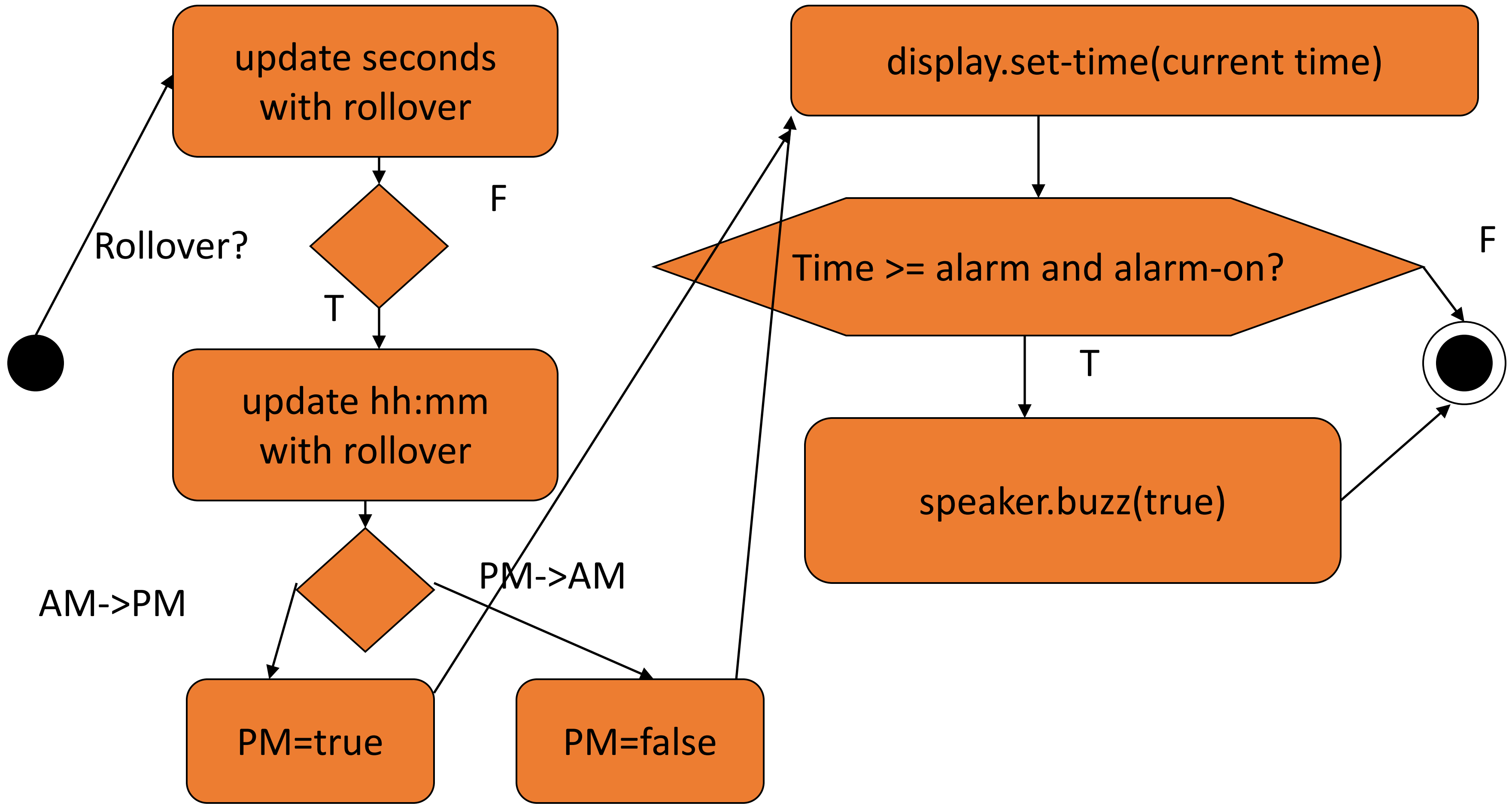
Mechanism class


Mechanism

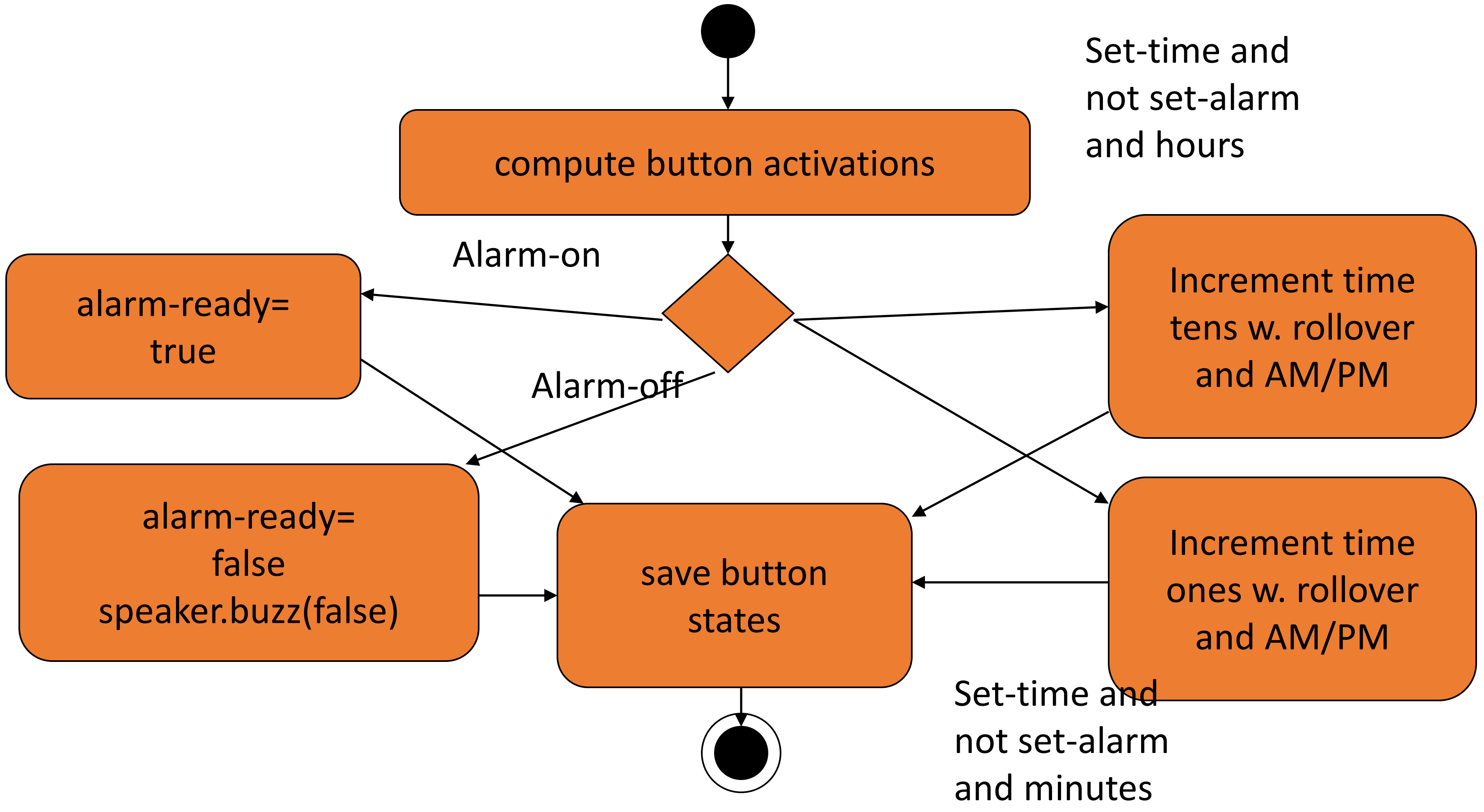
Seconds: integer
PM: boolean
tens-hours, ones-hours: boolean
tens-minutes, ones-minutes: boolean
alarm-ready: boolean
alarm-tens-hours, alarm-ones-hours: boolean
alarm-tens-minutes, alarm-ones-minutes: boolean

scan-keyboard()
update-time()

Update-time behavior


update seconds with rollover
Rollover?
F
T
update hh:mm with rollover
AM->PM
PM->AM
PM=true
PM=false
display.set-time(current time)
Time >= alarm and alarm-on?
F
T
speaker.buzz(true)

Scan-keyboard behavior


compute button activations
Set-time and not set-alarm and hours
Alarm-on
alarm-ready= true
Increment time tens w. rollover and AM/PM
Alarm-off
alarm-ready= false speaker.buzz(false)
save button states
Increment time ones w. rollover and AM/PM
Set-time and not set-alarm and minutes

System architecture

- Includes:
  - periodic behavior (clock);
  - aperiodic behavior (buttons, buzzer activation).
- Two major software components:
  - interrupt-driven routine updates time;
  - foreground program deals with buttons, commands.

Interrupt-driven routine

- Timer probably can’t handle one-minute interrupt interval.
- Use software variable to convert interrupt frequency to seconds.

Foreground program

- Operates as while loop:

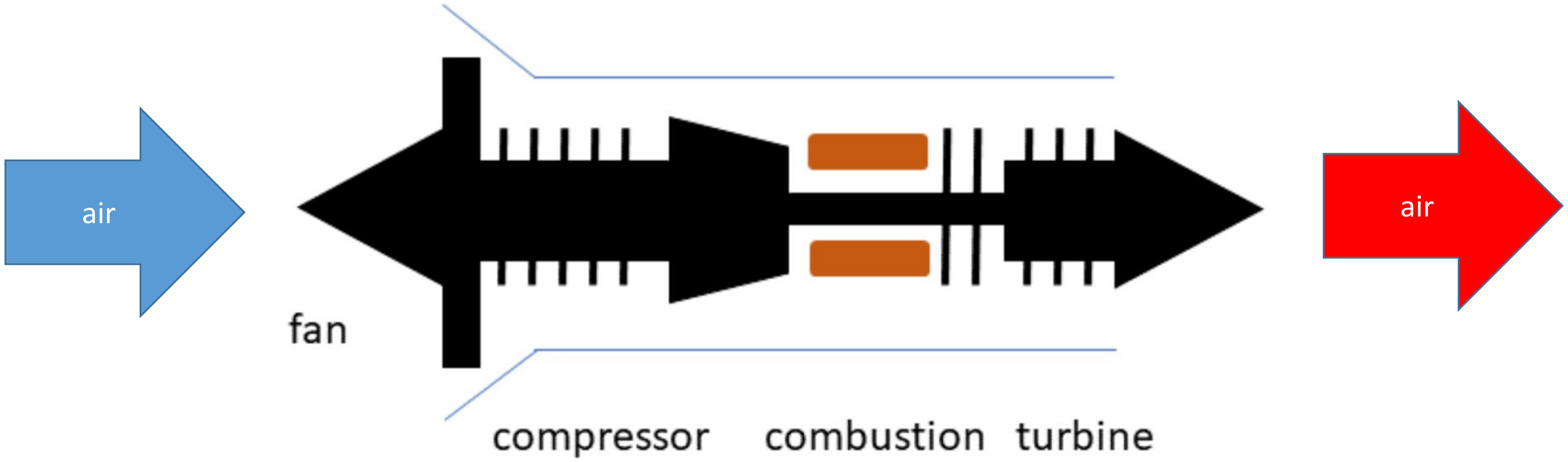
```
while (TRUE) {
        read_buttons(button_values);
        process_command(button_values);
        check_alarm();
}
```

Testing

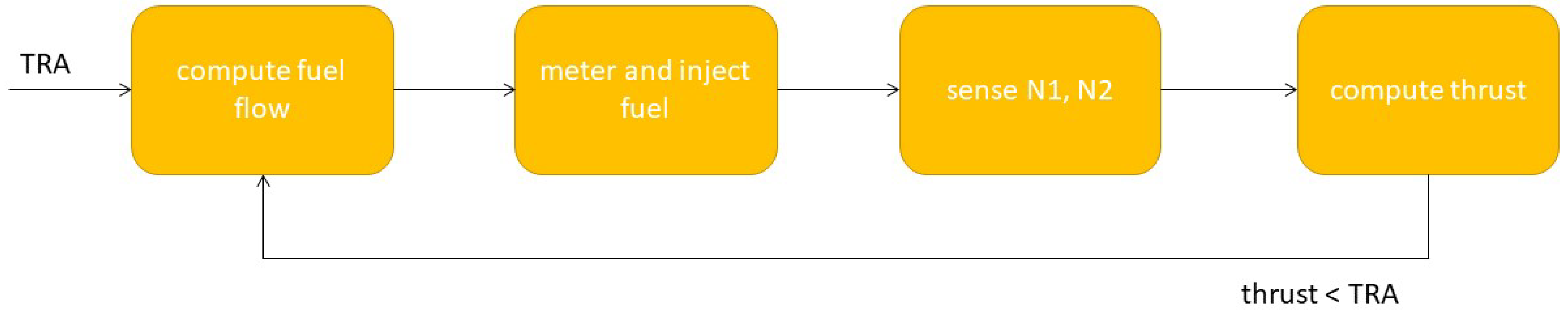
- Component testing:
    - test interrupt code on the platform;
    - can test foreground program using a mock-up.
- System testing:
    - relatively few components to integrate;
    - check clock accuracy;
    - check recognition of buttons, buzzer, etc.

Computing Platforms

- Example: jet engine controller


air
fan
compressor
combustion
turbine
air

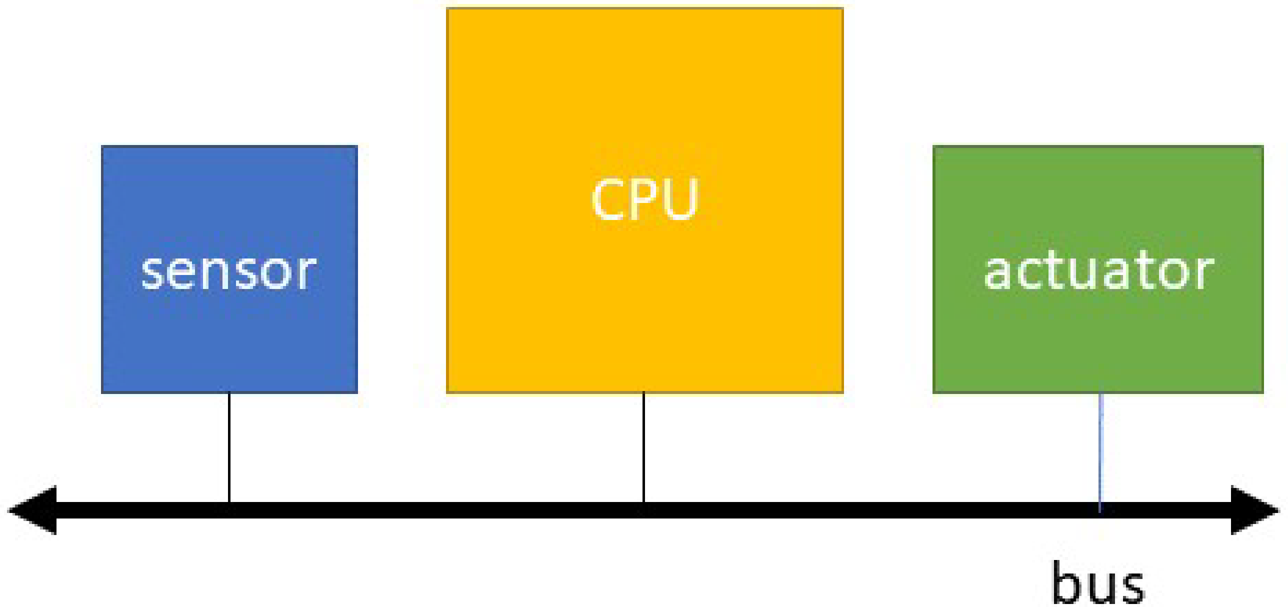
Full authority digital electronic control (FADEC)



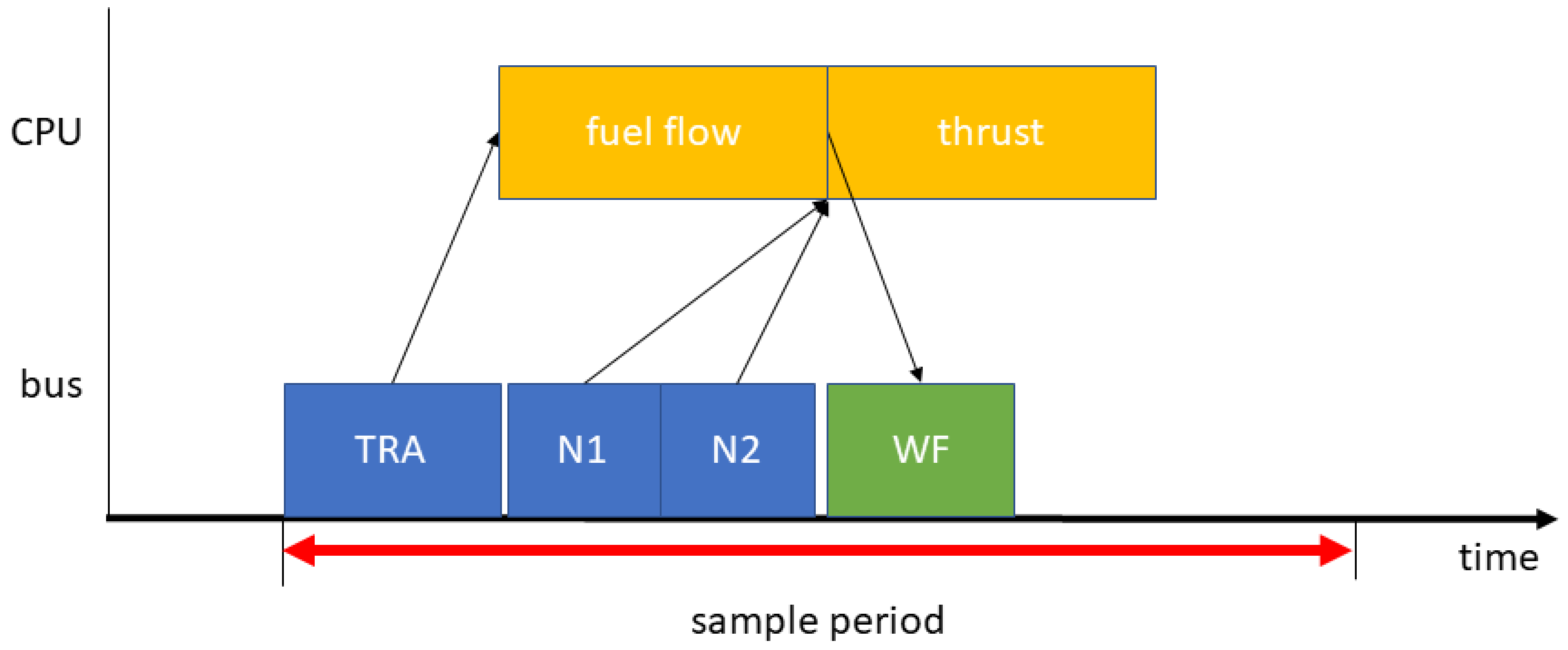
Control algorithm

- Inputs: throttle resolver angle (TRA), low pressure shaft speed (N1), high pressure shaft speed (N2).
- Sample in 10-100 Hz range.
- Operation:
  - The throttle resolver angle is the pilot’s command input to the jet engine.
  - Fuel flow WF is computed based on TRA and the current computed thrust; that fuel flow is sent to the fuel system.
  - The shaft speeds N1 and N2 are sensed.
  - Thrust is computed from shaft speeds. The result is stored for the next control cycle.

FADEC platform



Computation and I/O schedule


CPU
bus
fuel flow
thrust
TRA
N1
N2
WF
time
sample period