Bus-Based Computer Systems

- ⌘CPU bus, I/O devices, and interfacing
- ⌘CPU system as a framework
- ⌘System level performance
- ⌘Development and debugging
- ⌘An alarm clock design

Bus-Based Computer Systems

- Busses.
- Memory devices.
- I/O devices:
  - serial links
  - timers and counters
  - keyboards
  - displays
  - analog I/O

System architectures

- Architectures and components:
  - software;
  - hardware.
- Some software is very hardware-dependent.

Hardware platform architecture

Contains several elements:

- CPU;
- bus;
- memory;
- I/O devices: networking, sensors, actuators, etc.

How big/fast much each one be?

Software architecture

Functional description must be broken into pieces:

- division among people;
- conceptual organization;
- performance;
- testability;
- maintenance.

HW/SW architectures

Hardware and software are intimately related:

- software doesn't run without hardware;
- how much hardware you need is determined by the software requirements:
  - speed;
  - memory.

Evaluation boards

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

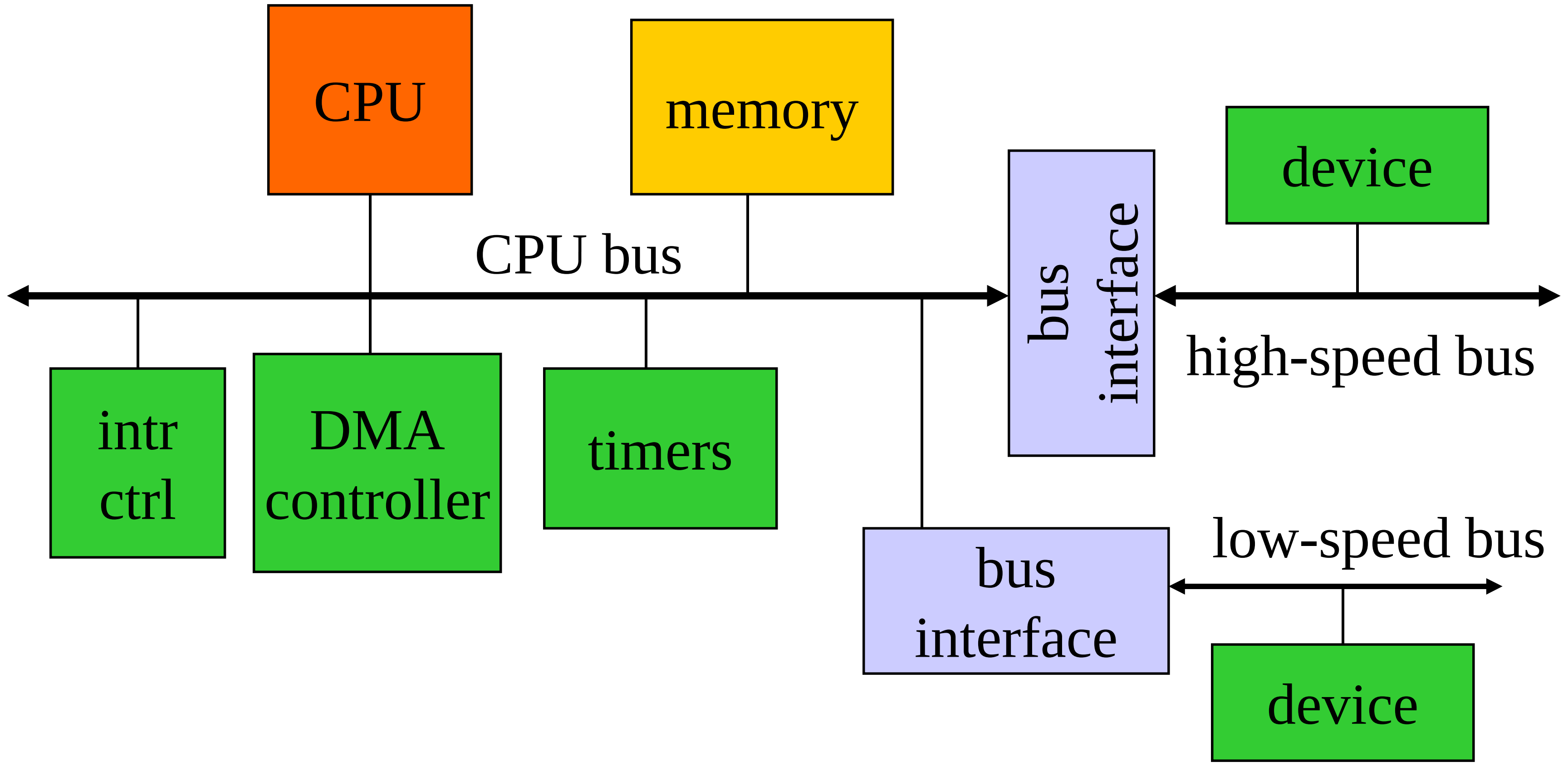
Adding logic to a board

- Programmable logic devices (PLDs) provide low/medium density logic.
- Field-programmable gate arrays (FPGAs) provide more logic and multi-level logic.
- Application-specific integrated circuits (ASICs) are manufactured for a single purpose.

The PC as a platform

- Advantages:
  - cheap and easy to get;
  - rich and familiar software environment.
- Disadvantages:
  - requires a lot of hardware resources;
  - not well-adapted to real-time.

Typical PC hardware platform



Typical busses

- PCI: standard for high-speed interfacing
  - 33 or 66 MHz.
  - PCI Express (PCIe): serial link.
    - 4 data wires per lane,
    - V1.x: 250 MB/s per lane
    - V2.0: 500 MB/s per lane
    - V3.0: 1GB/s per lane

- USB (Universal Serial Bus), Firewire (IEEE 1394): relatively low-cost serial interface with high speed.

Software elements

- IBM PC uses BIOS (Basic I/O System) to implement low-level functions:
  - boot-up;
  - minimal device drivers.
- BIOS has become a generic term for the lowest-level system software.

Example: StrongARM

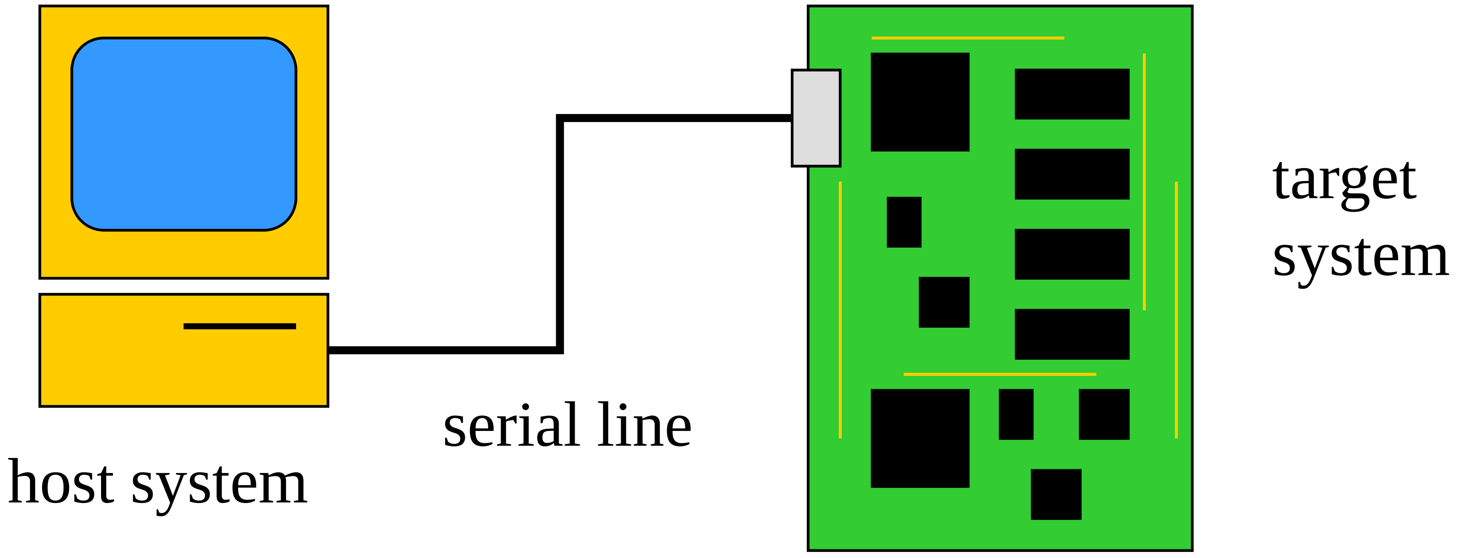
- StrongARM system includes:
  - CPU chip (3.686 MHz clock)
  - system control module (32.768 kHz clock).
    - Real-time clock;
    - operating system timer
    - general-purpose I/O;
    - interrupt controller;
    - power manager controller;
    - reset controller.

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.

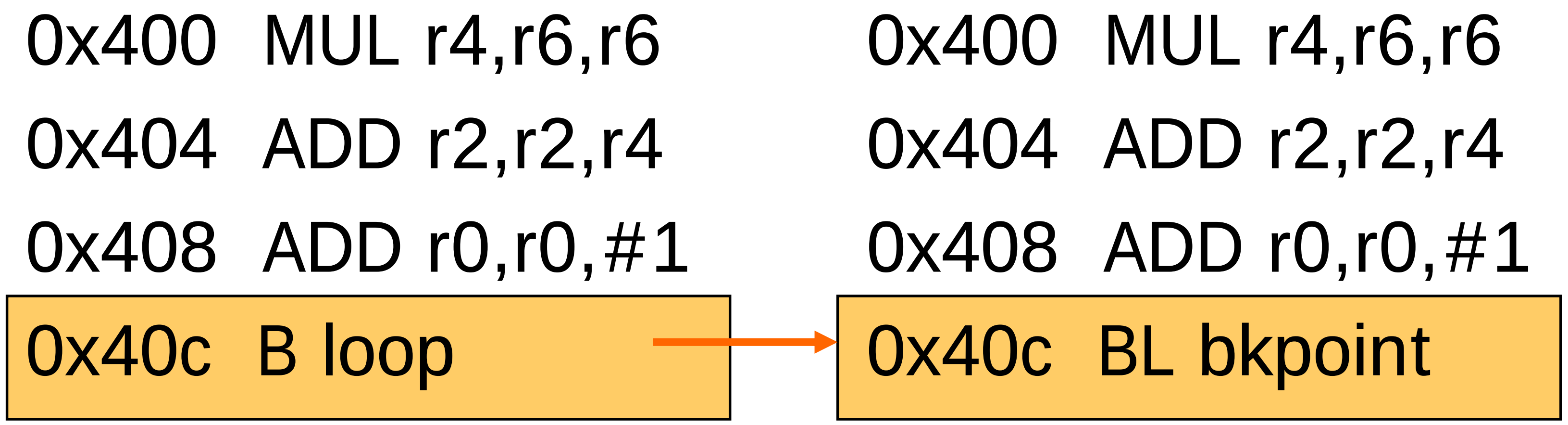
Software debuggers

- 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
```

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

uninstrumented code

code with breakpoint

Breakpoint handler actions

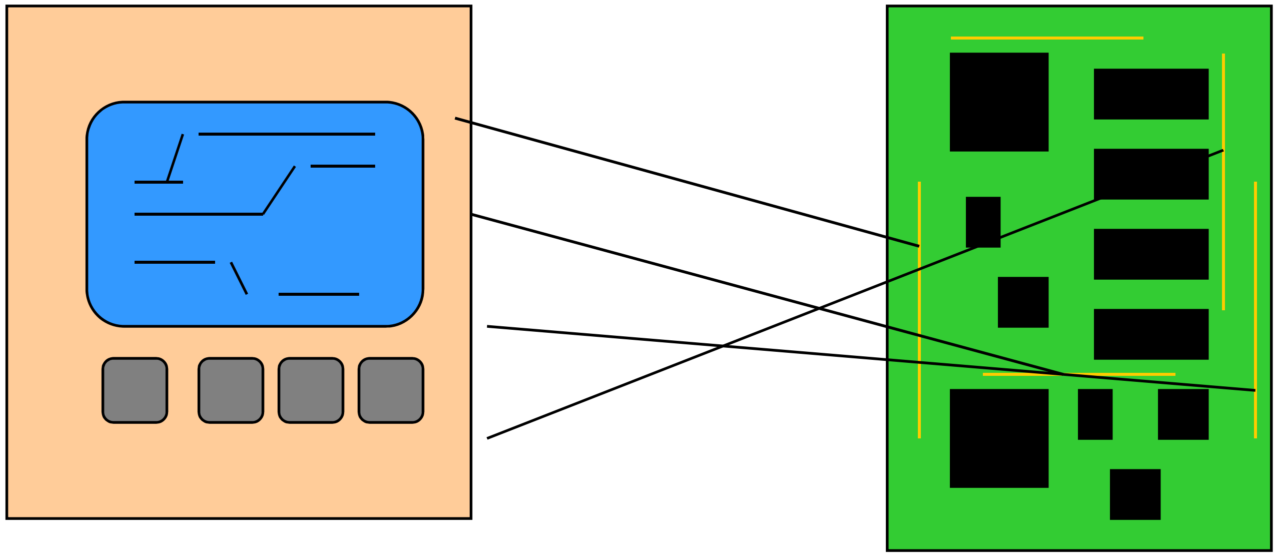
- Save registers.
- Allow user to examine machine.
- Before returning, restore system state.
  - Safest way to execute the instruction is to replace it and execute in place.
  - Put another breakpoint after the replaced breakpoint to allow restoring the original breakpoint.

In-circuit emulators (ICE)

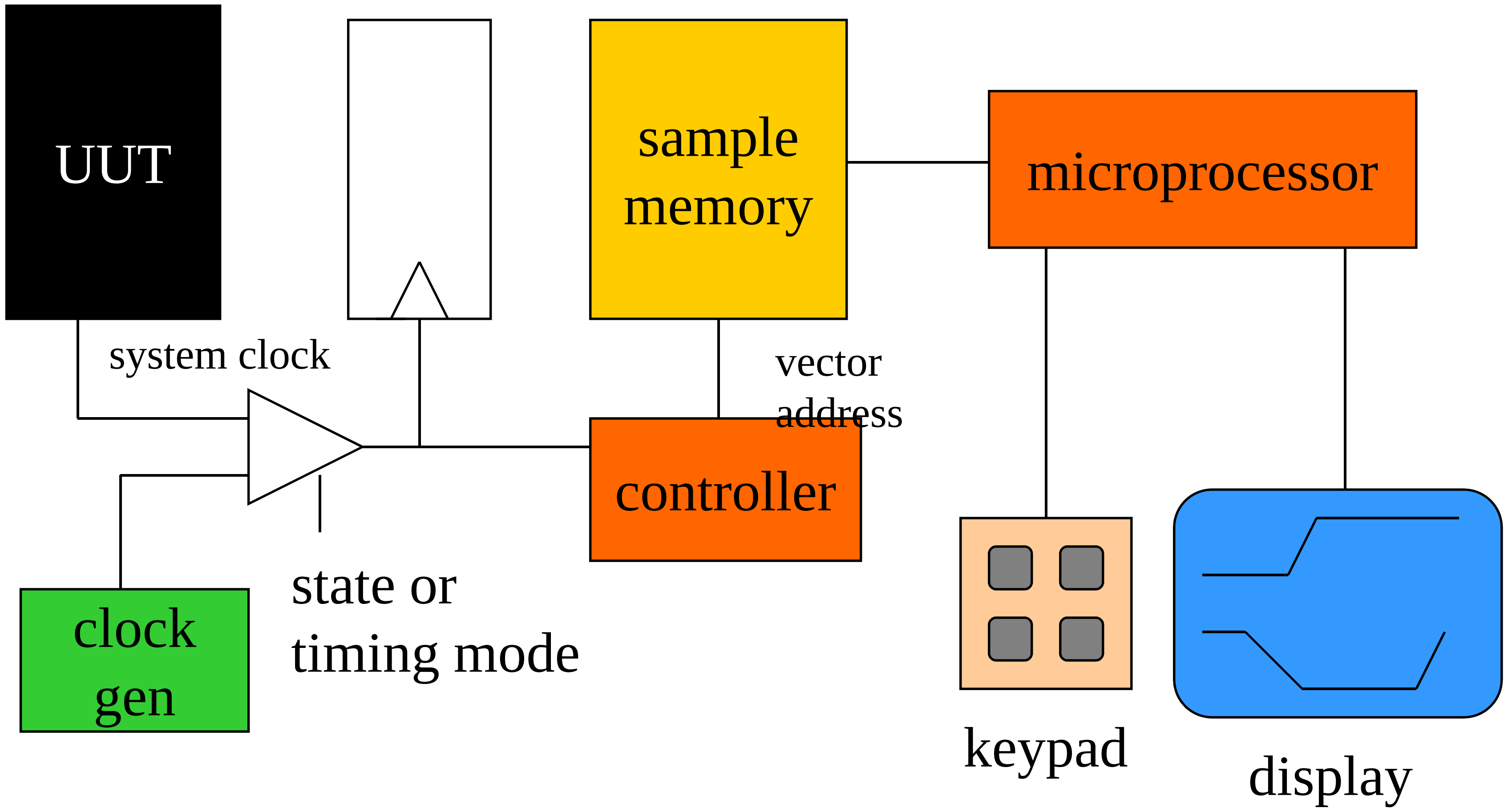
- A microprocessor in-circuit emulator is a specially-instrumented microprocessor.
- Allows you to stop execution, examine CPU state, modify registers.

Logic analyzers

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

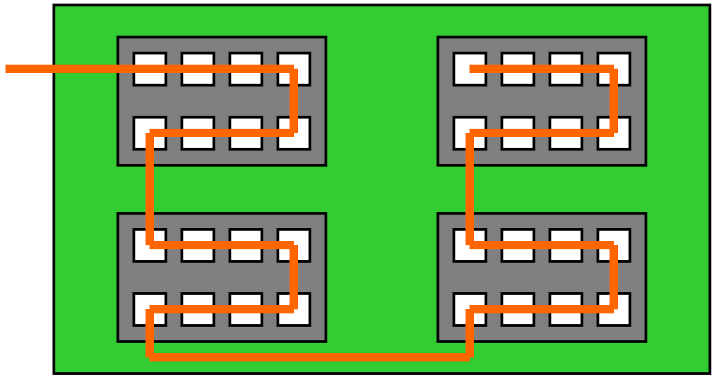


Logic analyzer architecture



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

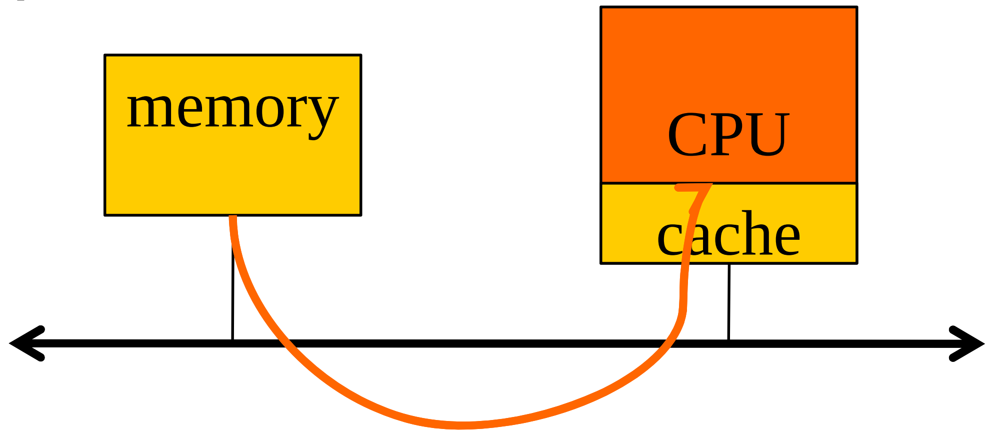
- ⌘Run on host system.
- ⌘Run on target system.
- ⌘Run in instruction-level simulator.
- ⌘Run on cycle-accurate simulator.
- ⌘Run in hardware/software co-simulation environment.

Debugging real-time code

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

System-level performance analysis

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



Bandwidth as performance

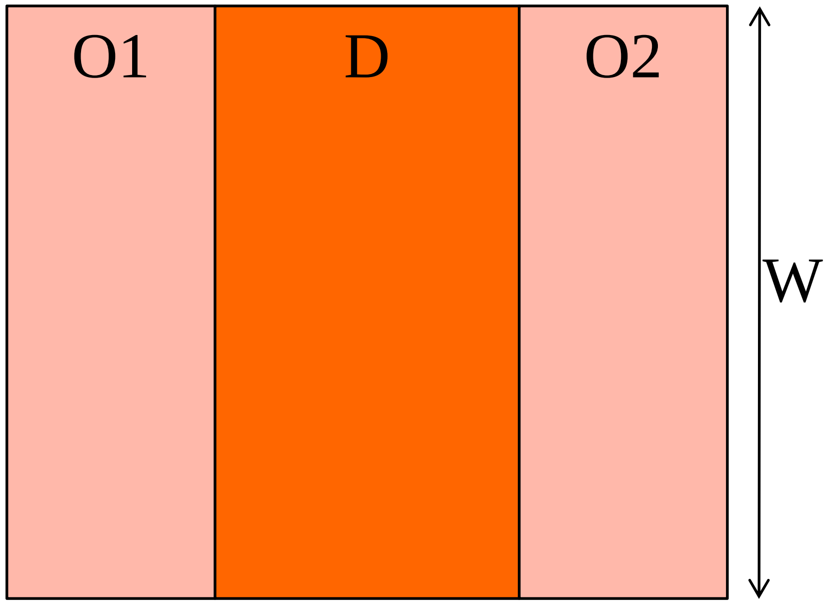
- Bandwidth applies to several components:
  - Memory.
  - Bus.
  - CPU fetches.
- Different parts of the system run at different clock rates.
- Different components may have different widths (bus, memory).

Bandwidth and data transfers

- Per video frame: 320 x 240 x 3 = 230,400 bytes.
  - Transfer in 1/30 sec.
- Transfer 1 byte/μsec, 0.23 sec per frame.
  - Too slow.
- Increase bandwidth:
  - Increase bus width.
  - Increase bus clock rate.

Bus bandwidth

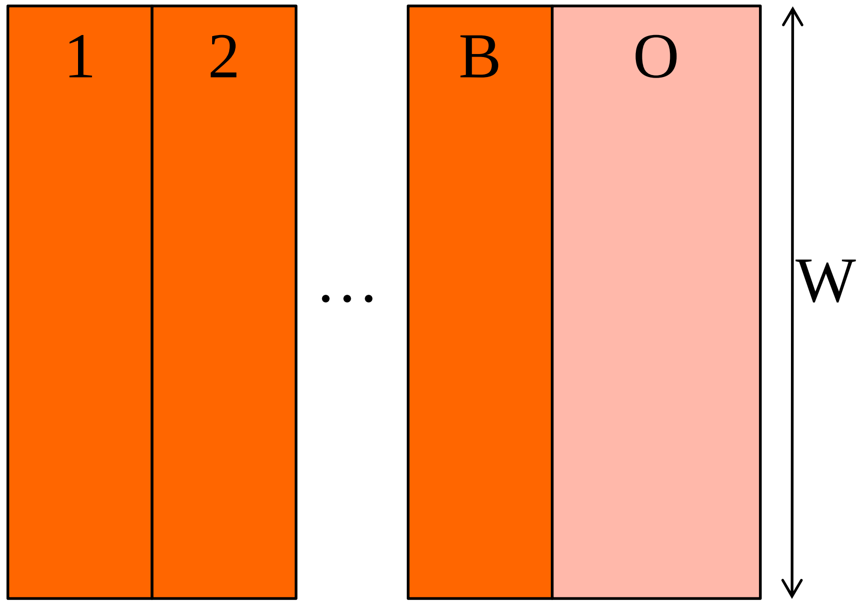
- T: # bus cycles.
- P: time/bus cycle.
- Total time for transfer:
  - t = TP.
- D: data payload length.
- O1 + O2 = overhead O.
  - Address, handshaking
- N bytes to be transferred
- Bus width: W bytes



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

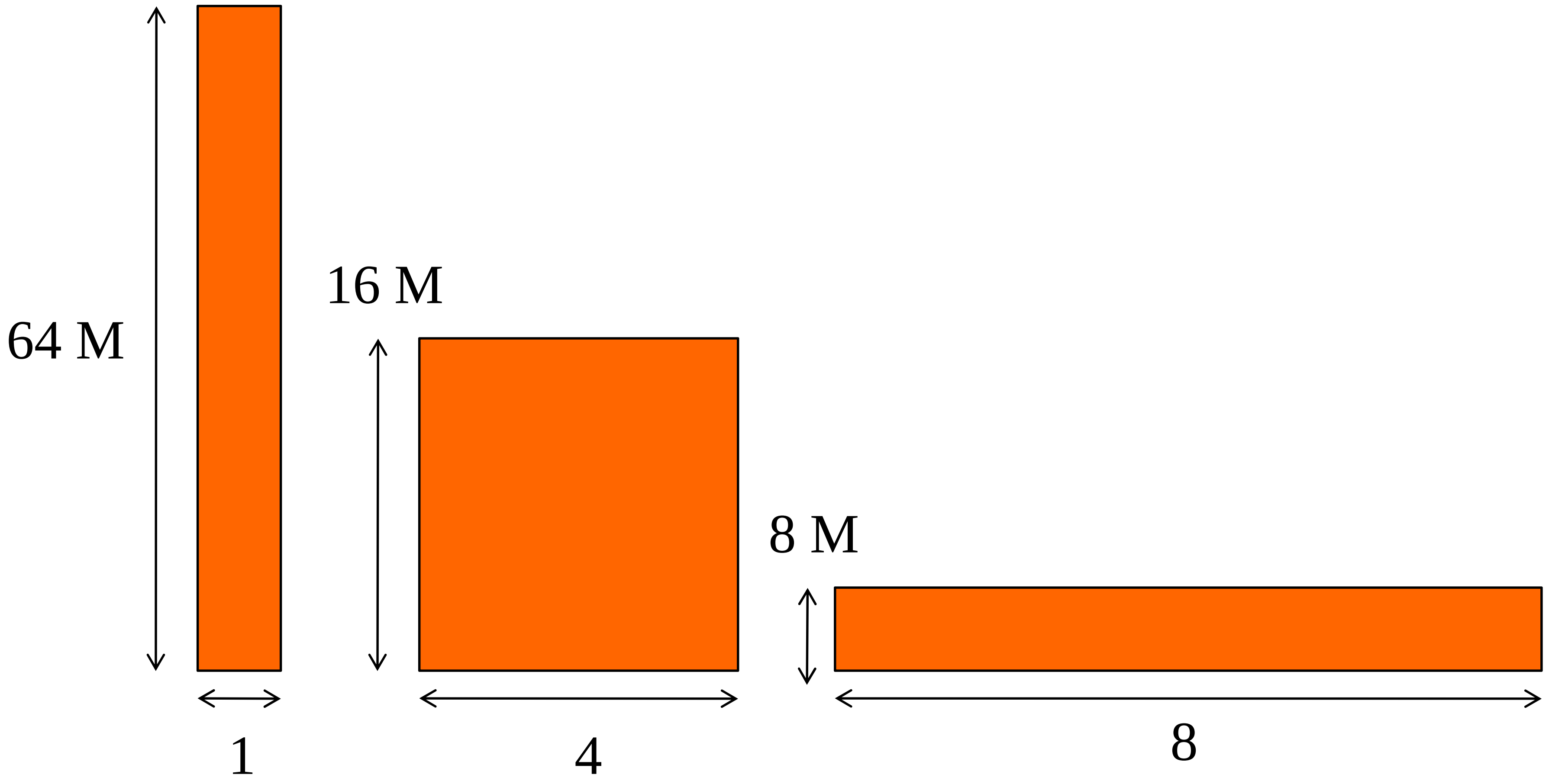
Bus burst transfer bandwidth

- T: # bus cycles.
- P: time/bus cycle.
- Total time for transfer:
  - t = TP.
- D: data payload length.
- O1 + O2 = overhead O.



$$T_{burst}(N) = (BD+O)N/(BW)$$

Memory aspect ratios

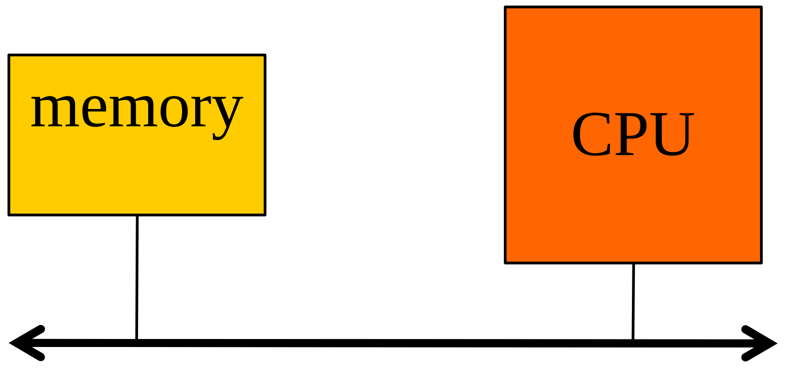


Memory access times

- Memory component access times comes from chip data sheet.
  - Page modes allow faster access for successive transfers on same page.
- What if data doesn't fit naturally into physical words:
- A pixel: RGB 24-bit
  - an access for 24-bit-wide memory
  - 3 accesses for 8-bit wide memory
  - how about 32-bit wide memory
    - waste one byte for each access
    - packing

Bus performance bottlenecks

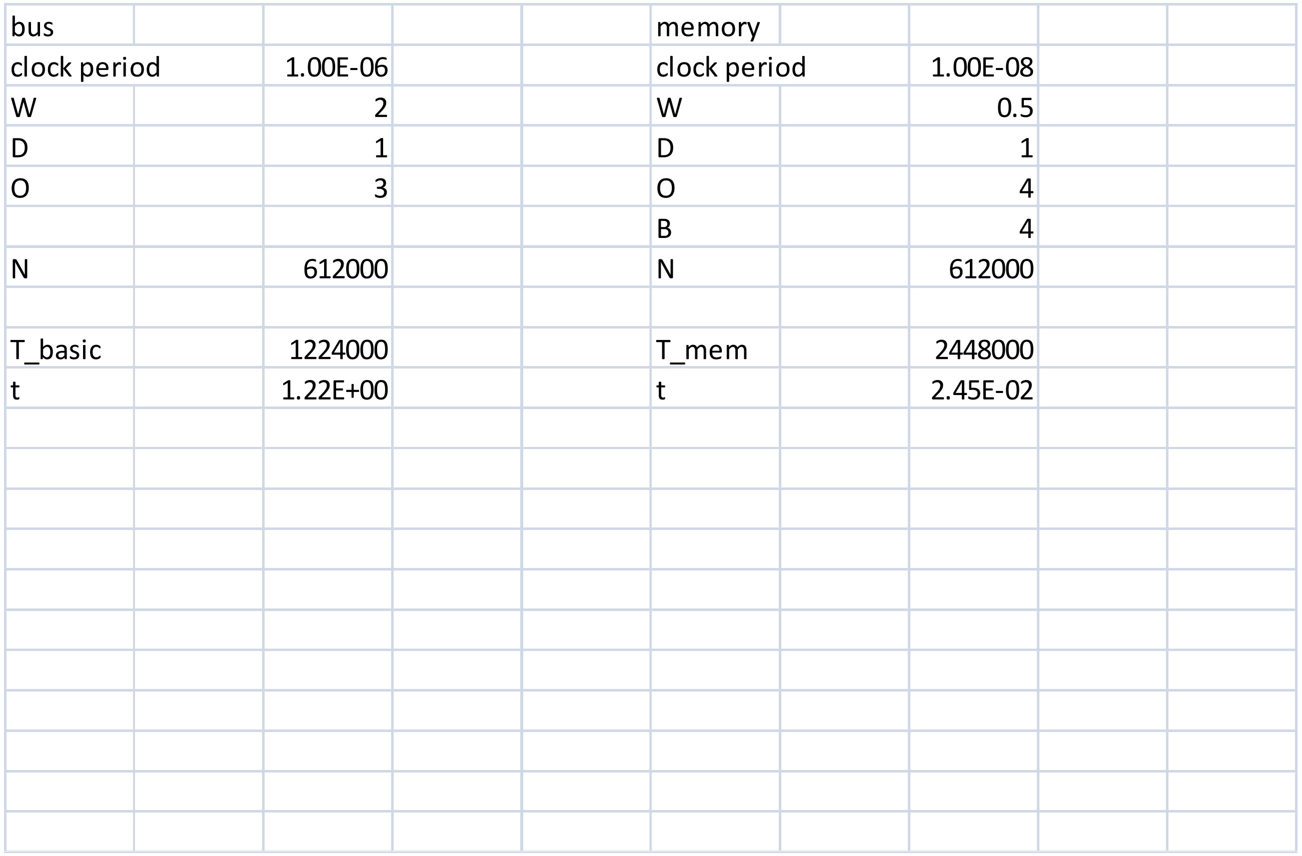
- Transfer 320 x 240 video frame @ 30 frames/sec = 612,000 bytes/sec.
- Is performance bottleneck bus or memory?



Bus performance bottlenecks, cont'd.

- Bus: assume 1 MHz bus, D=1, O=3:
  - T_{basic} = (1+3)612,000/2 = 1,224,000 cycles = 1.224 sec.
- Memory: try burst mode B=4, width w=0.5. (assume 10MHz)
  - T_{mem} = (4*1+4)612,000/(4*0.5) = 2,448,000 cycles = 0.2448 sec.

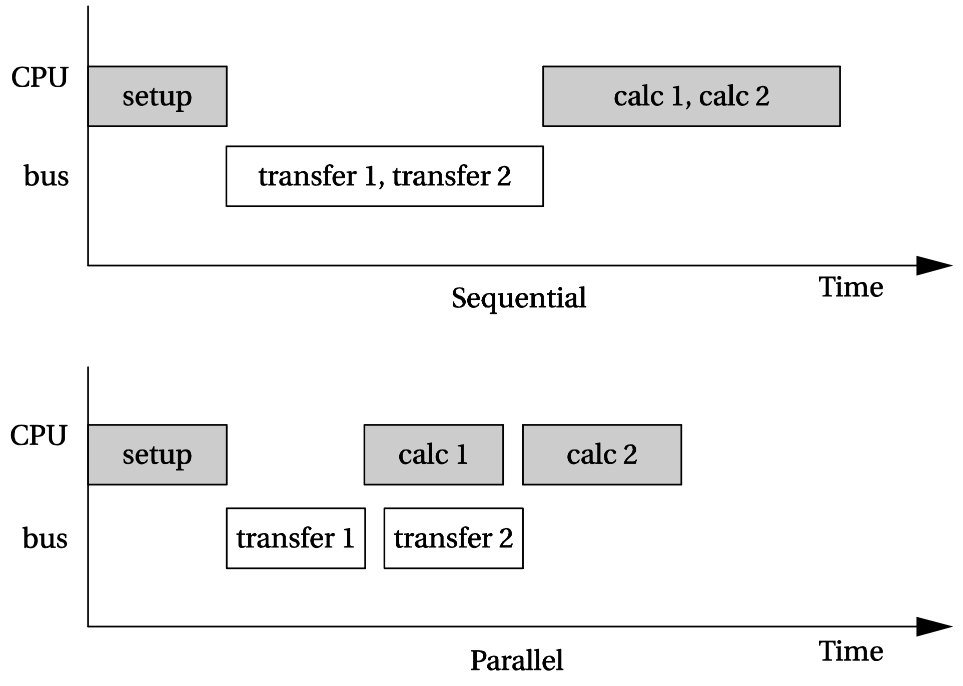
Performance spreadsheet



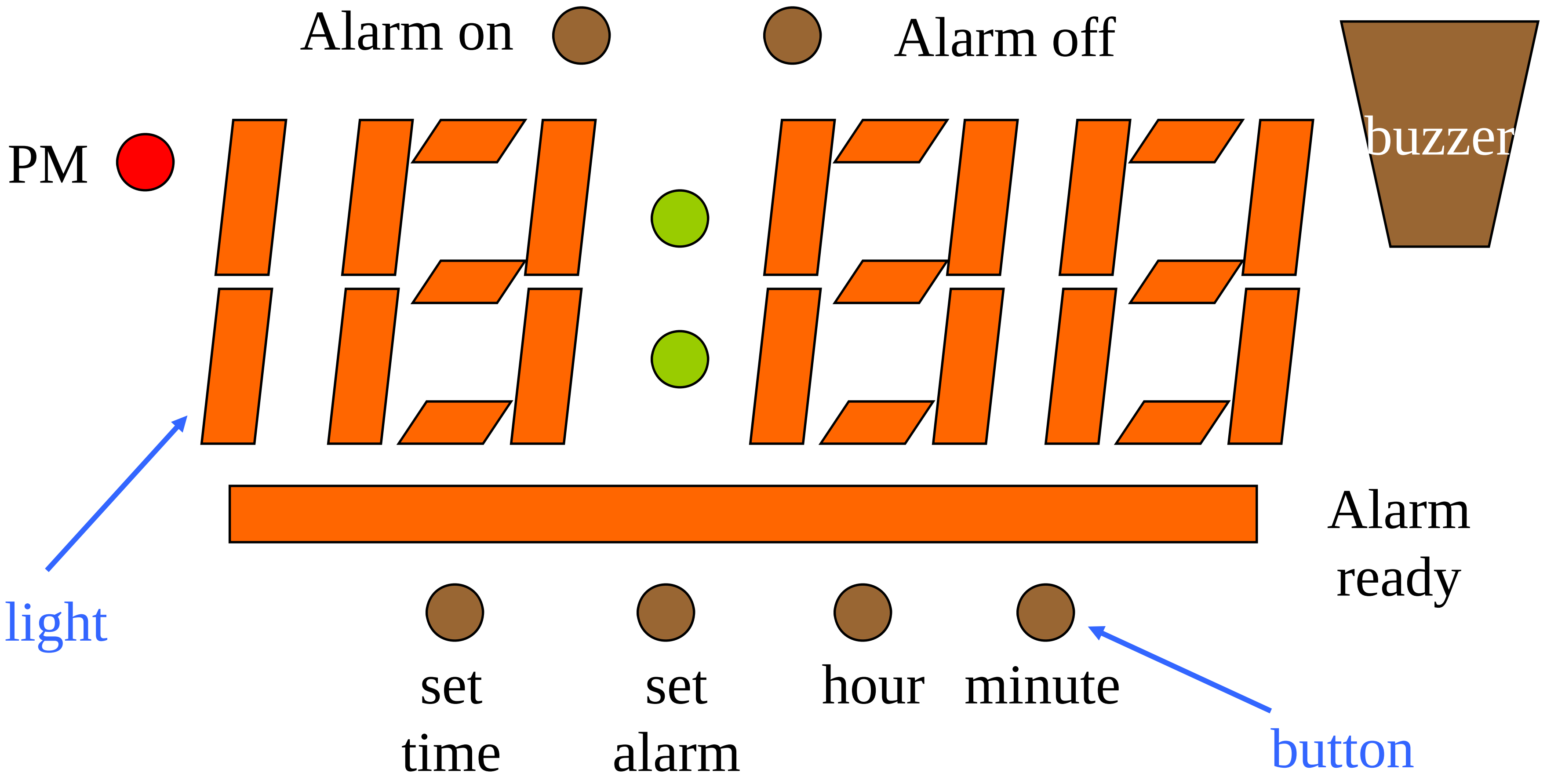
bus					memory				
clock period		1.00E-06			clock period		1.00E-08		
W		2			W		0.5		
D		1			D		1		
O		3			O		4		
					B		4		
N		612000			N		612000		
T_basic		1224000			T_mem		2448000		
t		1.22E+00			t		2.45E-02		

Parallelism

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



Alarm clock interface



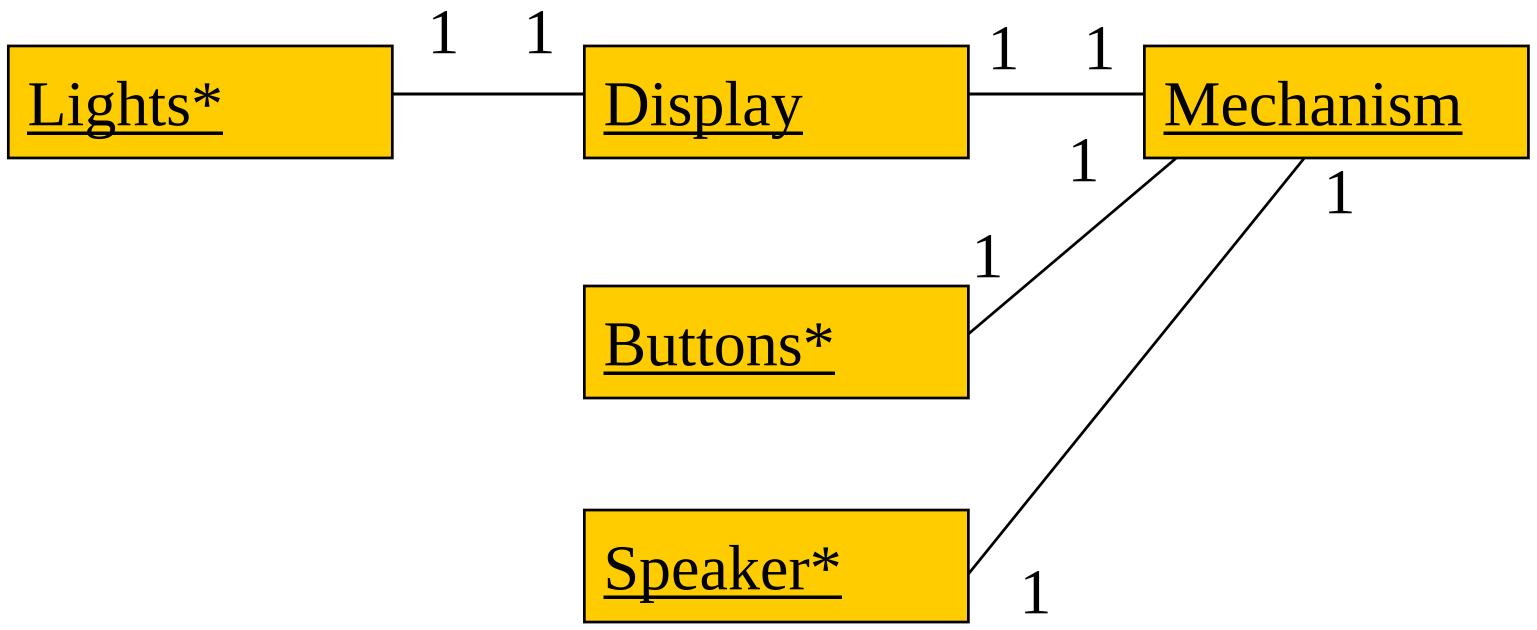
Operations

- 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.

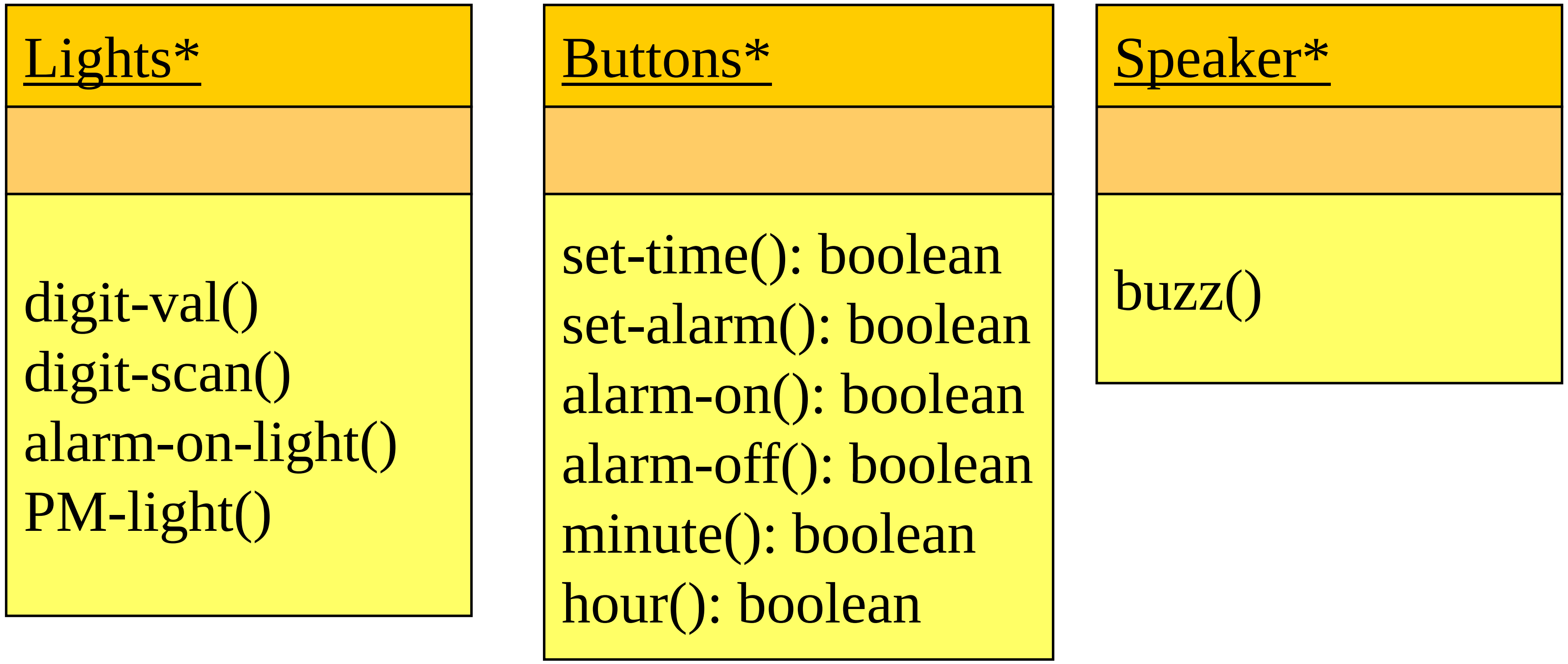
Alarm clock requirements

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

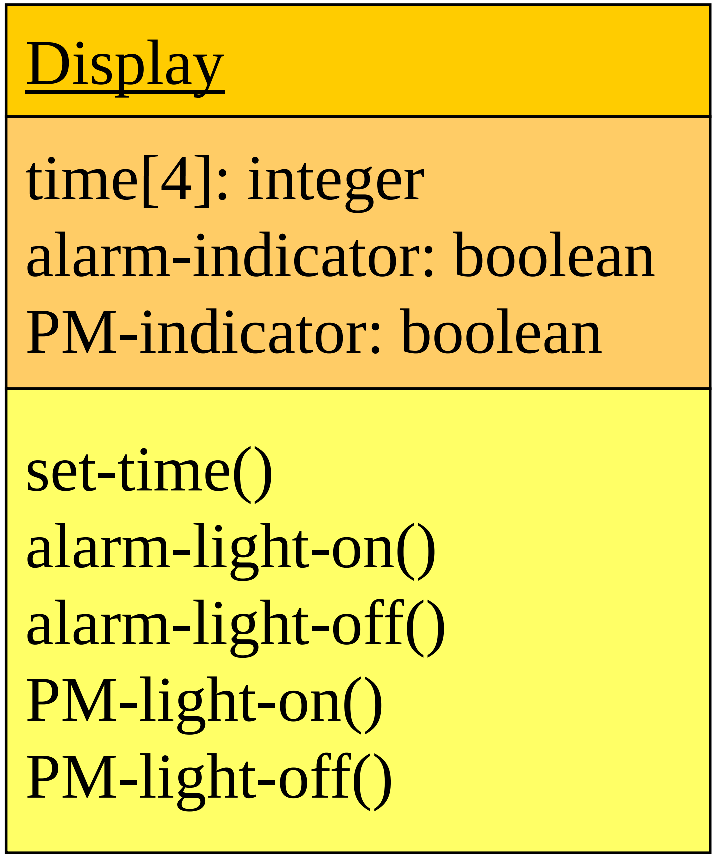
Alarm clock class diagram



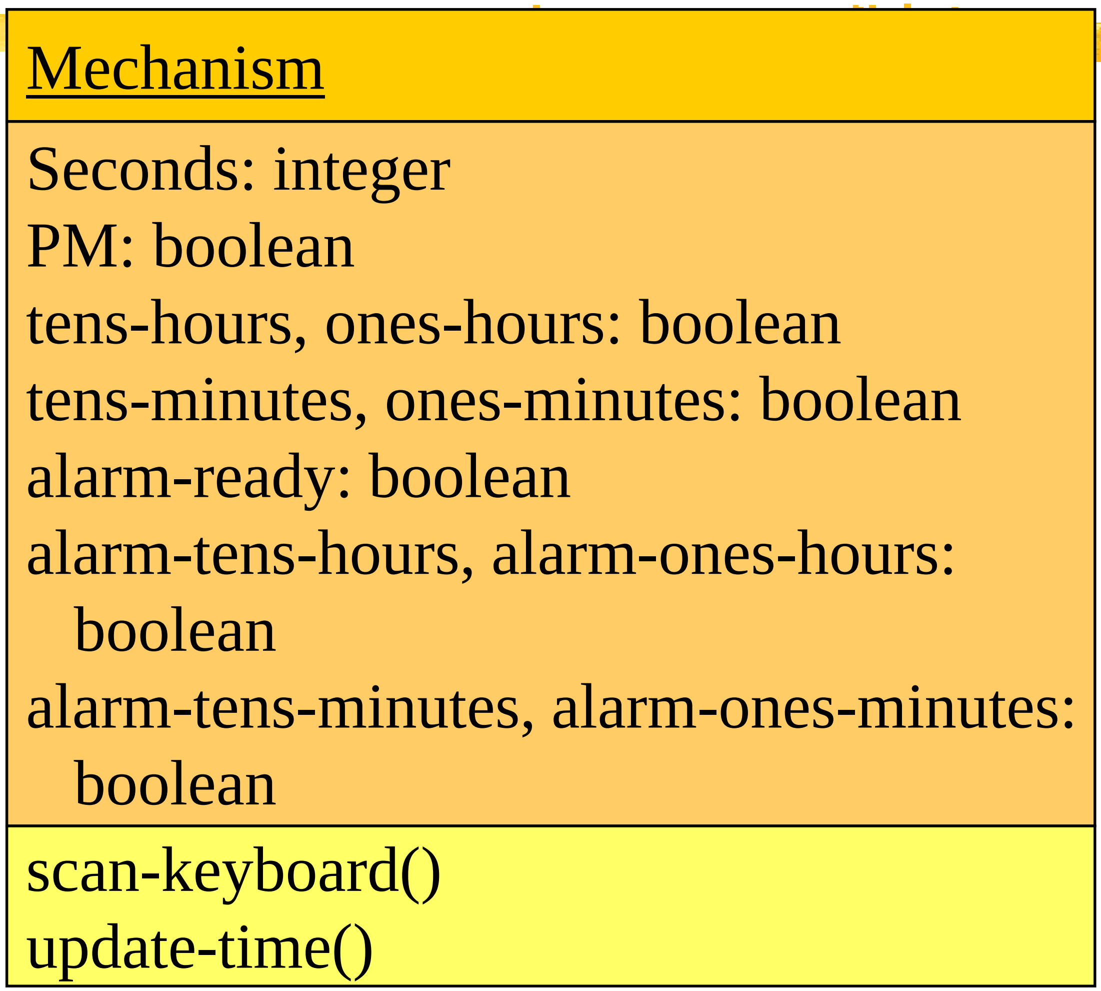
Alarm clock physical classes



Display class



Mechanism class

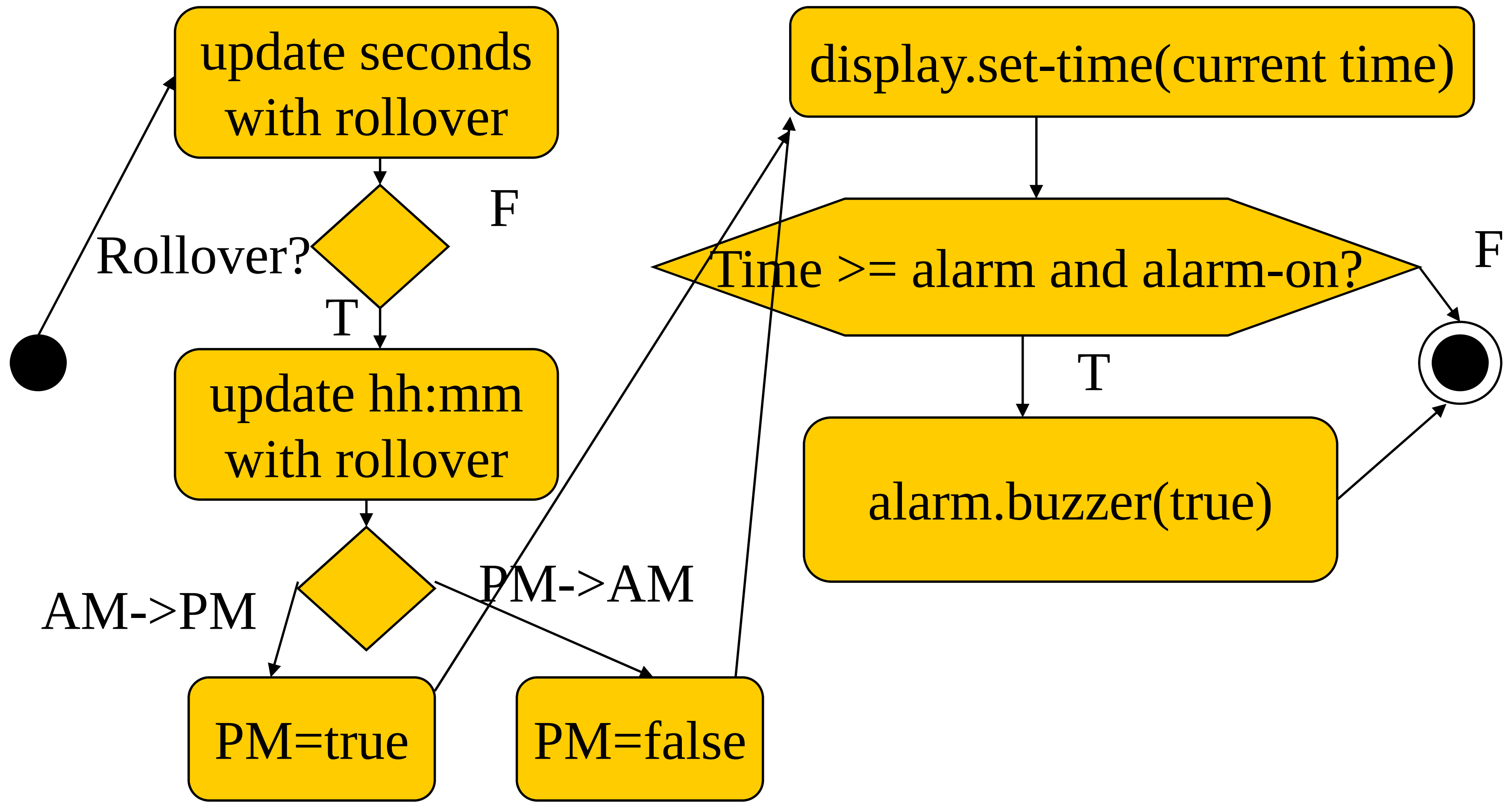


<u>Mechanism</u>

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

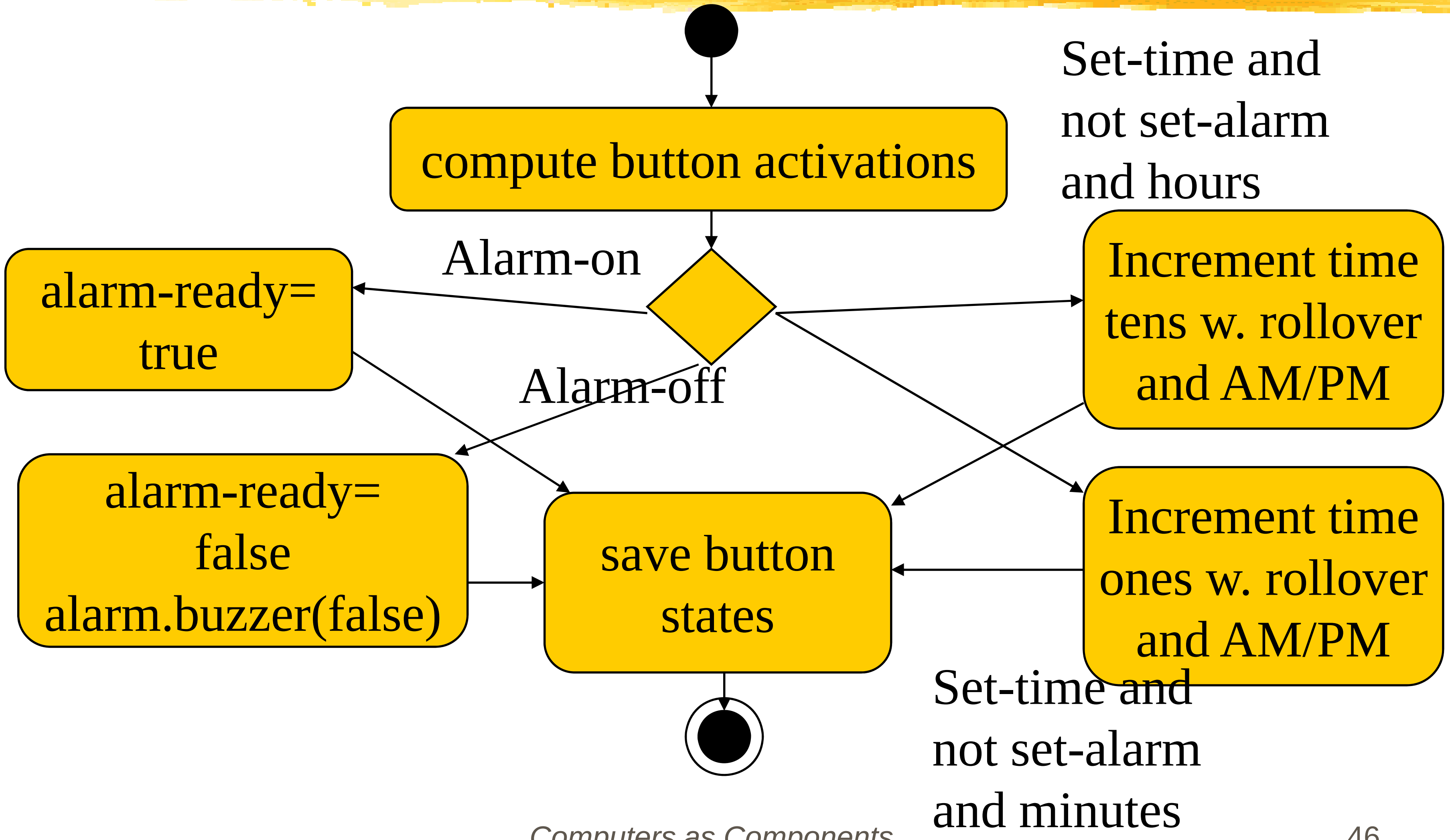
scan-keyboard()
update-time()

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
alarm.buzzer(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 alarm.buzzer(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.

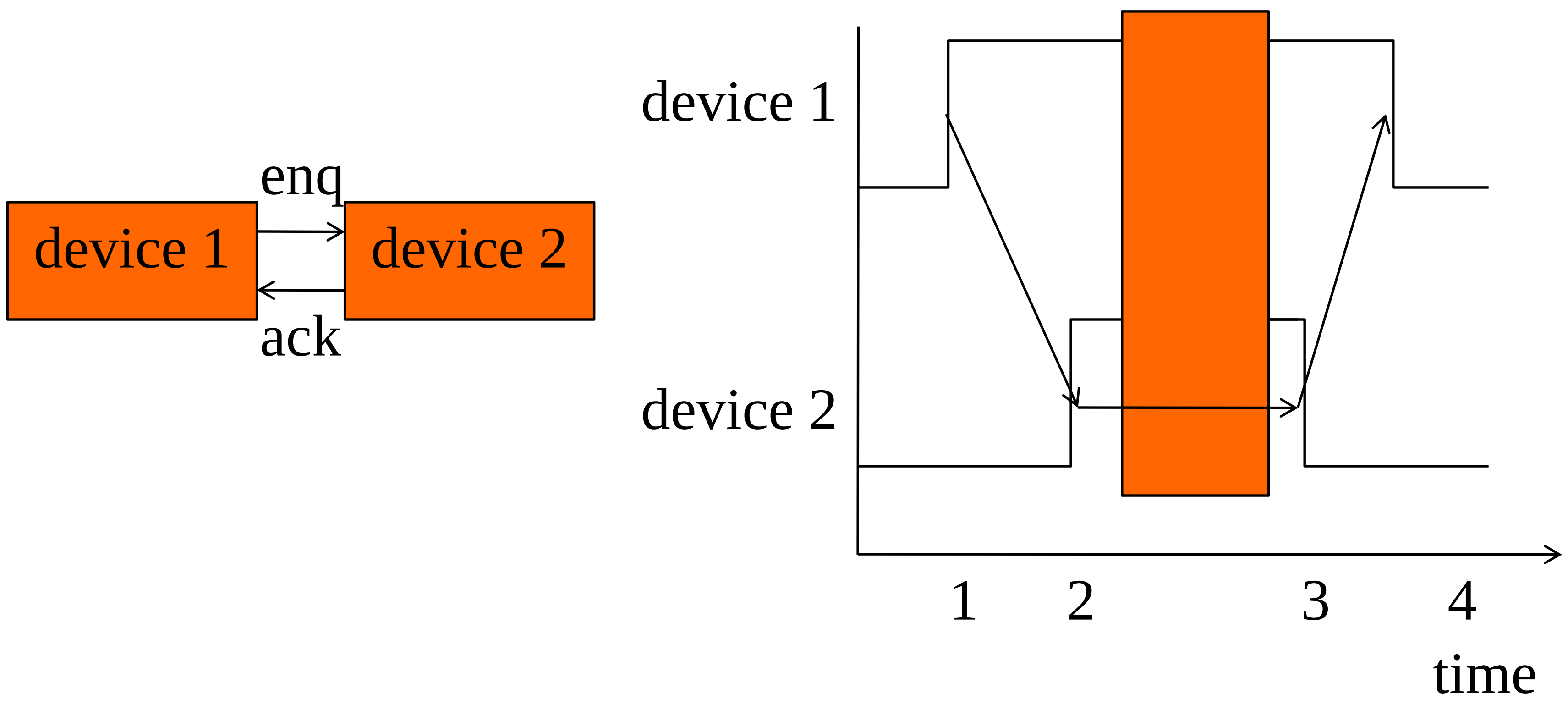
The CPU bus

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

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

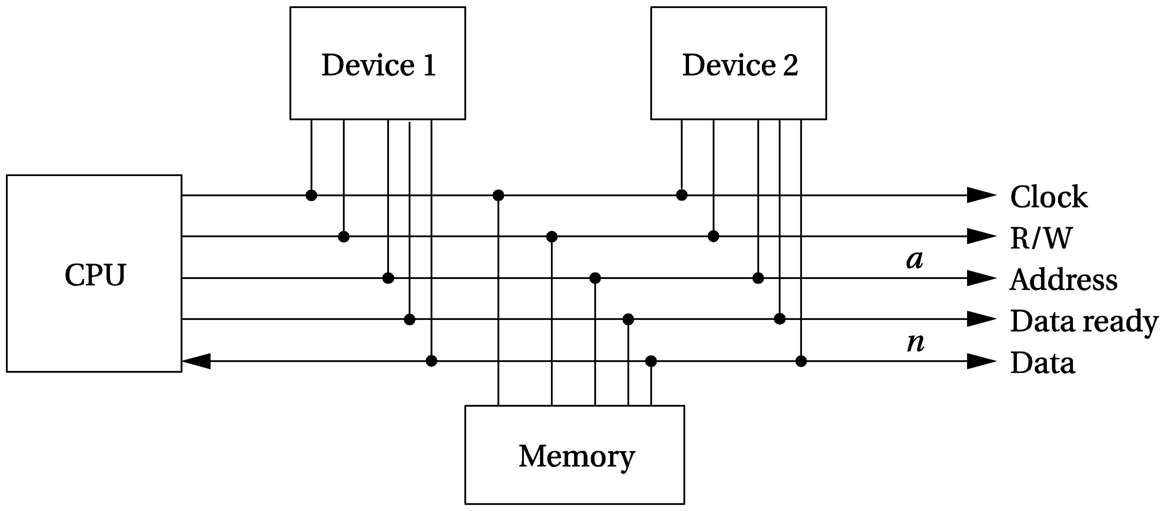


Four-cycle handshake

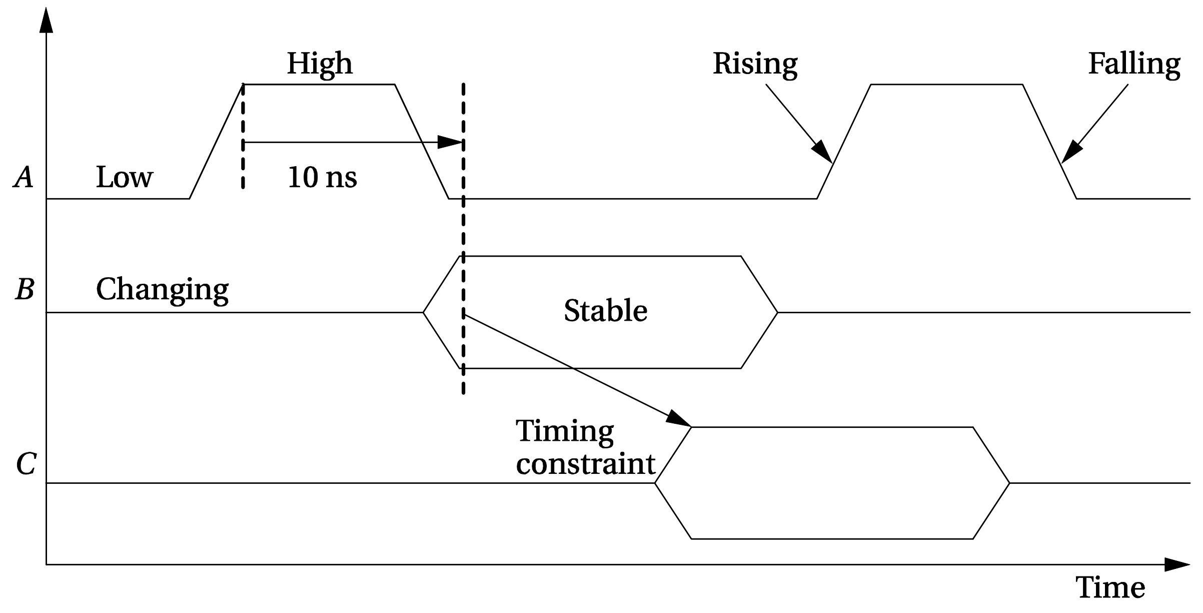
1. Device 1 raises enq.
2. Device 2 responds with ack.
3. Device 2 lowers ack once it has finished.
4. Device 1 lowers enq.

Microprocessor busses

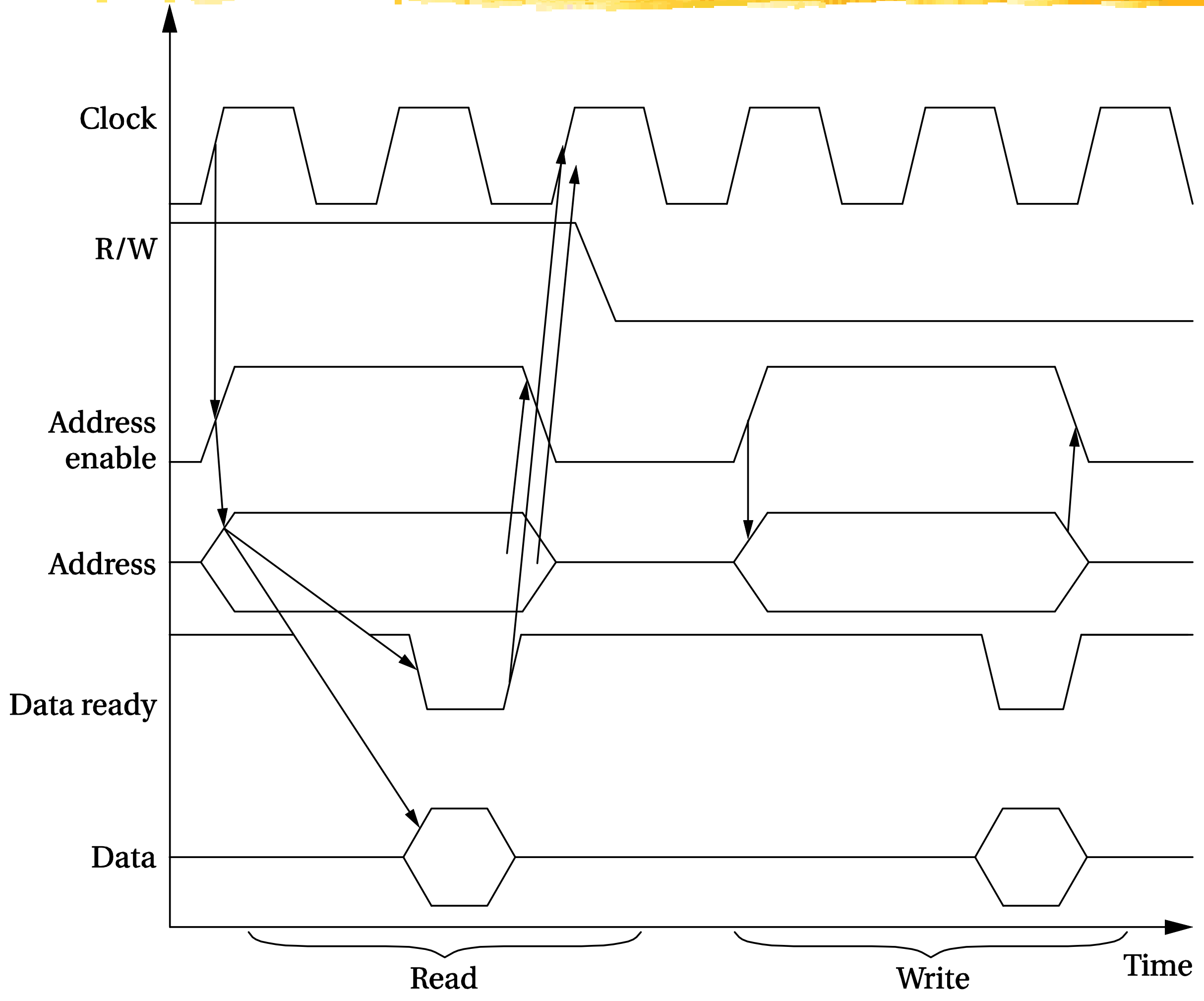
- Clock provides synchronization.
- R/W is true when reading (R/W' is false when reading).
- Address is a-bit bundle of address lines.
- Data is n-bit bundle of data lines.
- Data ready signals when n-bit data is ready.



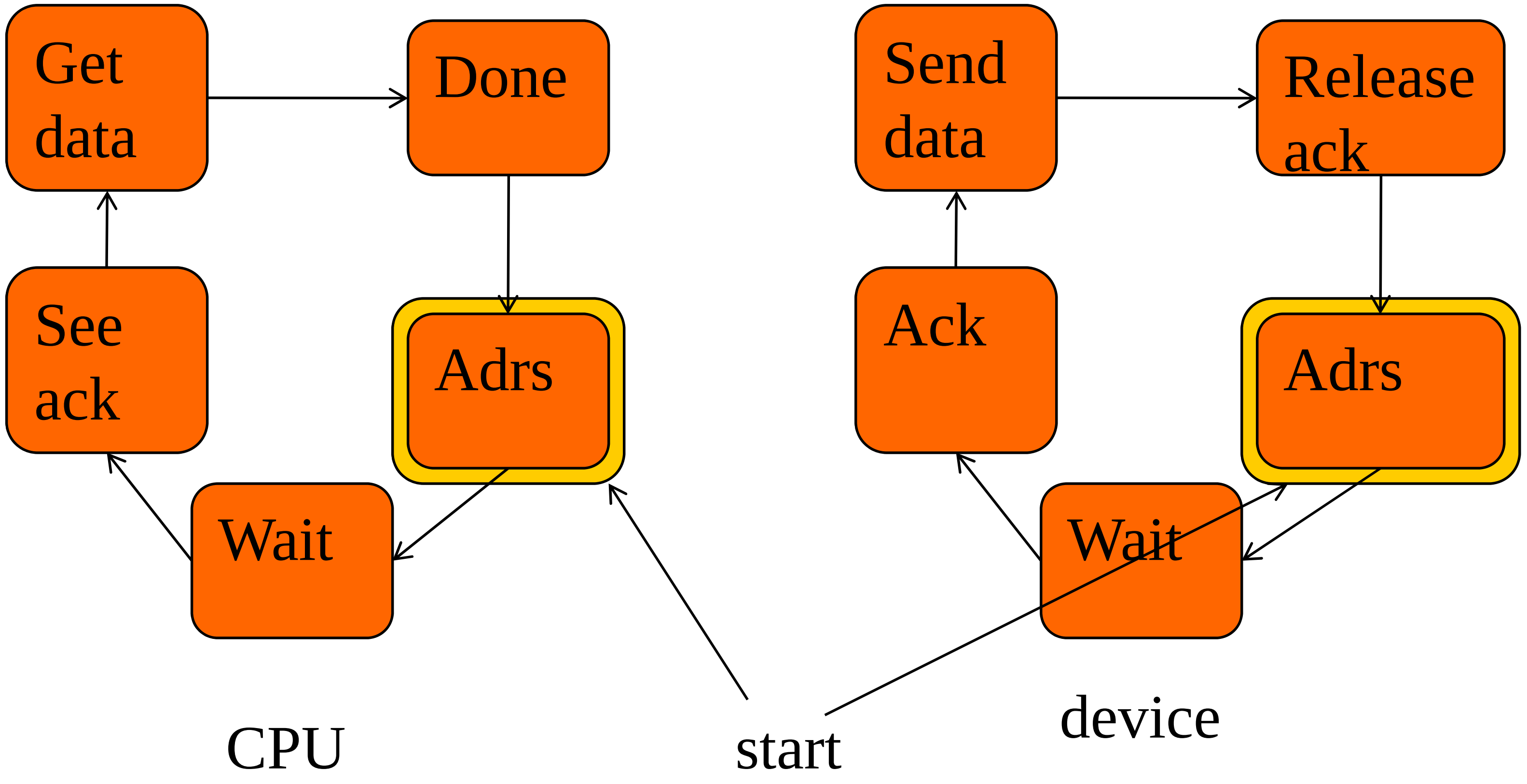
Timing diagrams



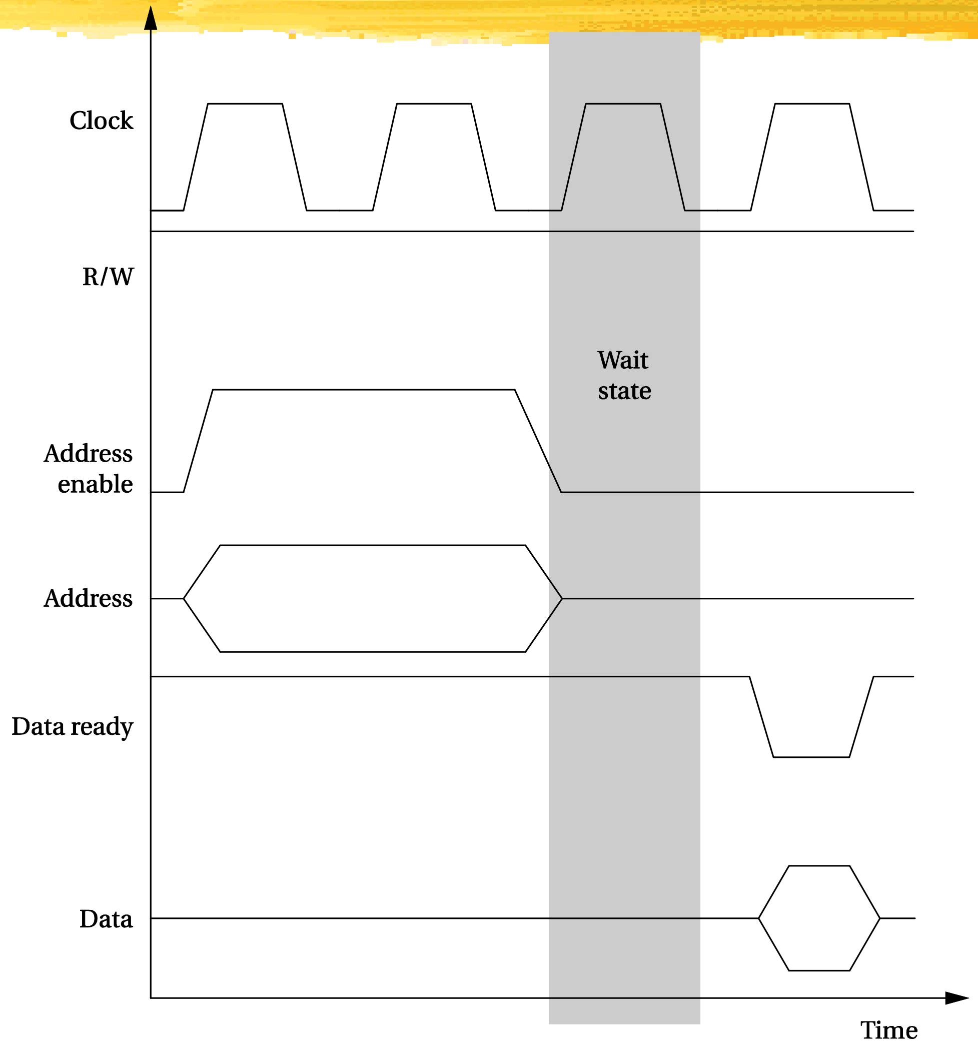
Bus read


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

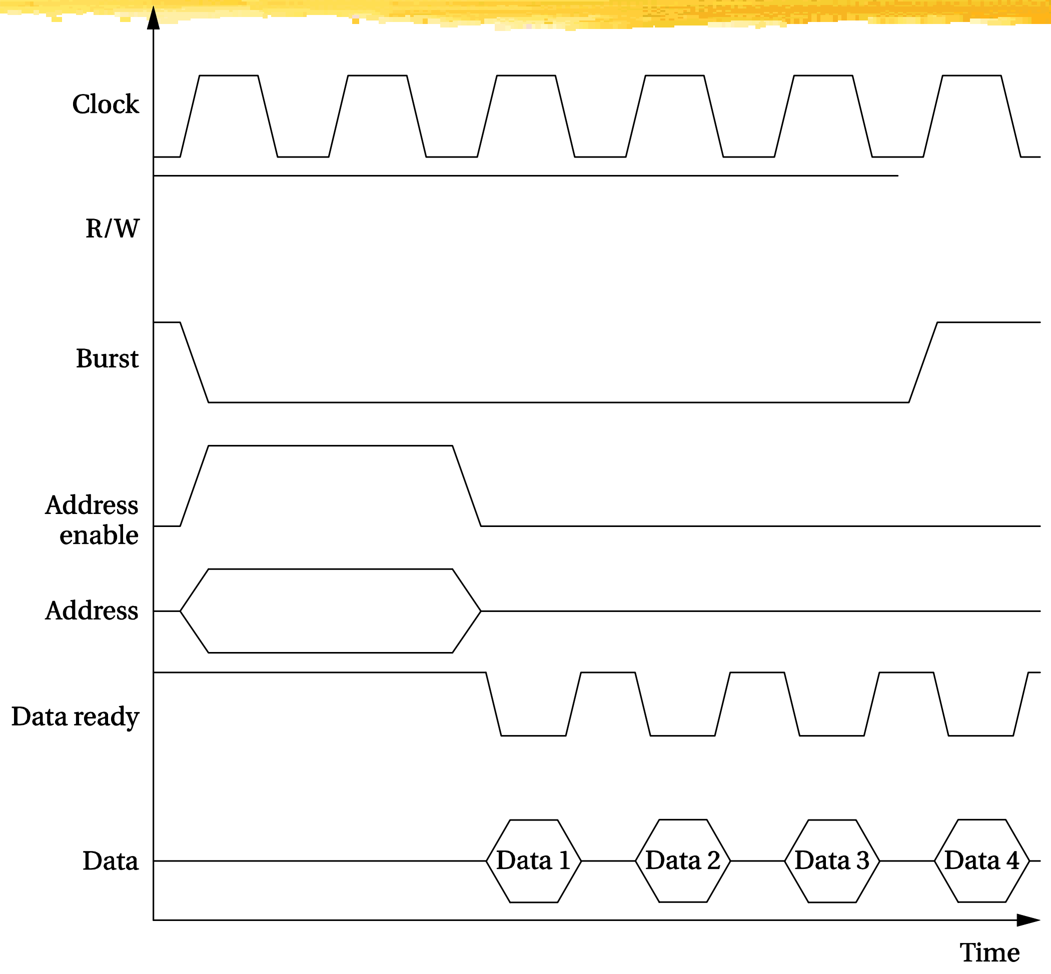
State diagrams for bus read



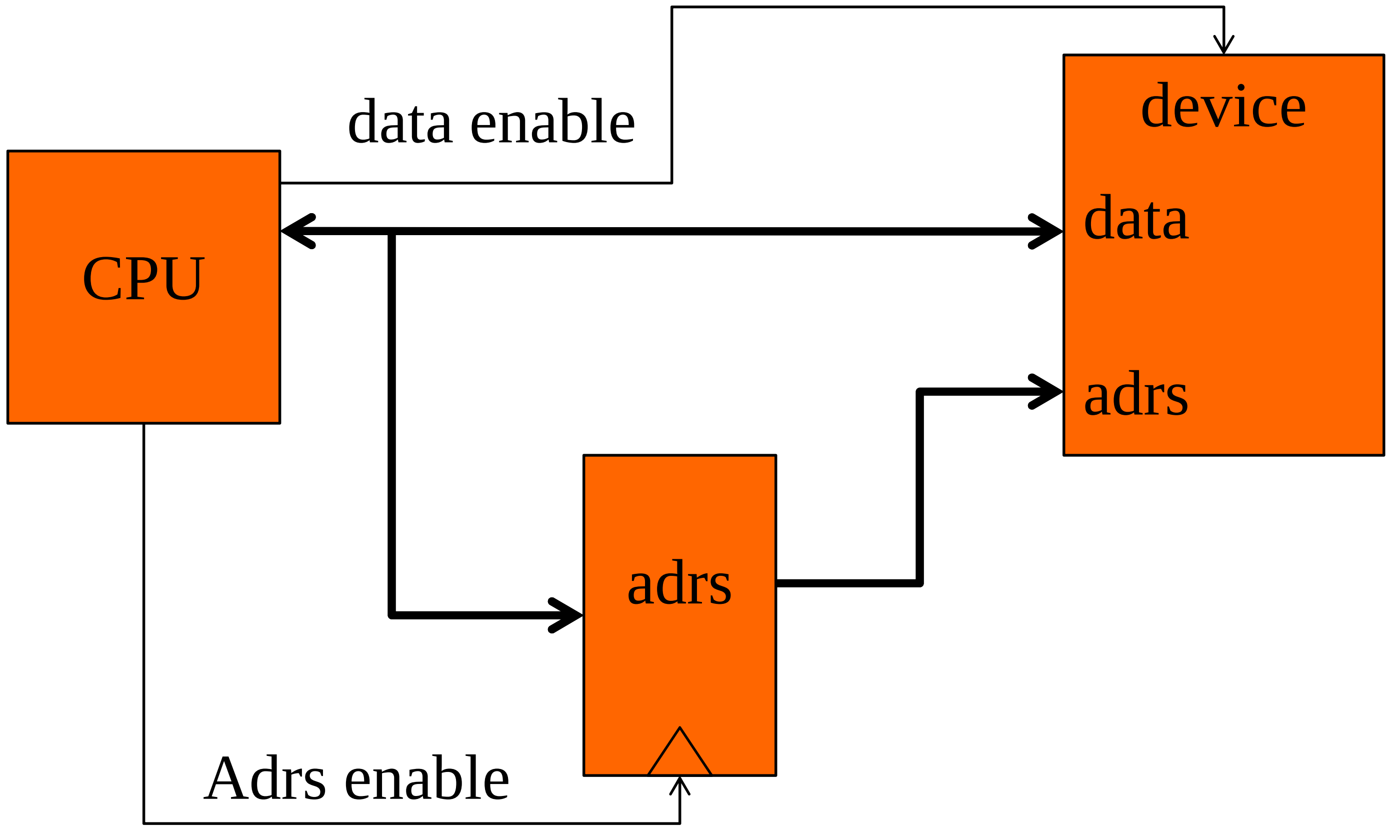
Bus wait state



Bus burst read

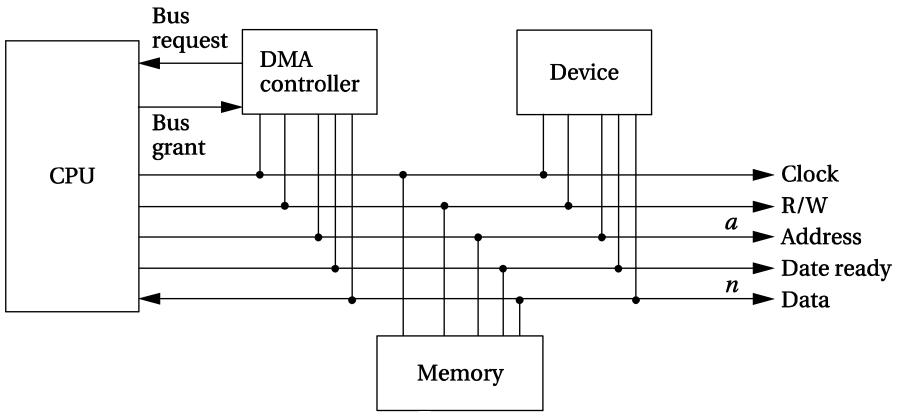


Bus multiplexing



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.

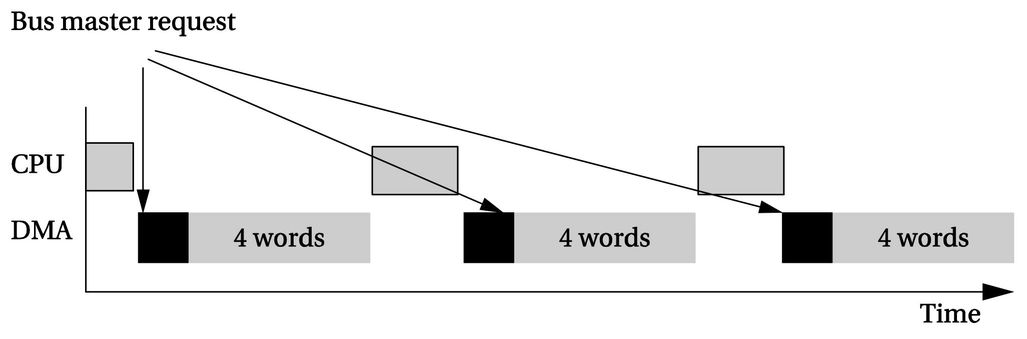


Bus mastership

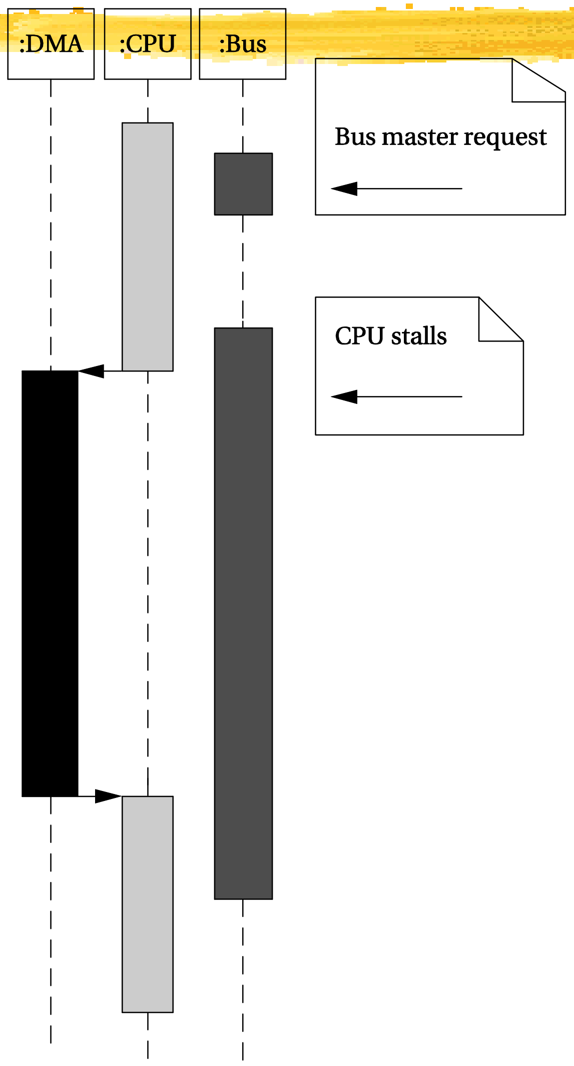
- By default, CPU is bus master and initiates transfers.
- DMA must become bus master to perform its work.
  - CPU can't use bus while DMA operates.
- Bus mastership protocol:
  - Bus request.
  - Bus grant.

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 nth bus cycle.

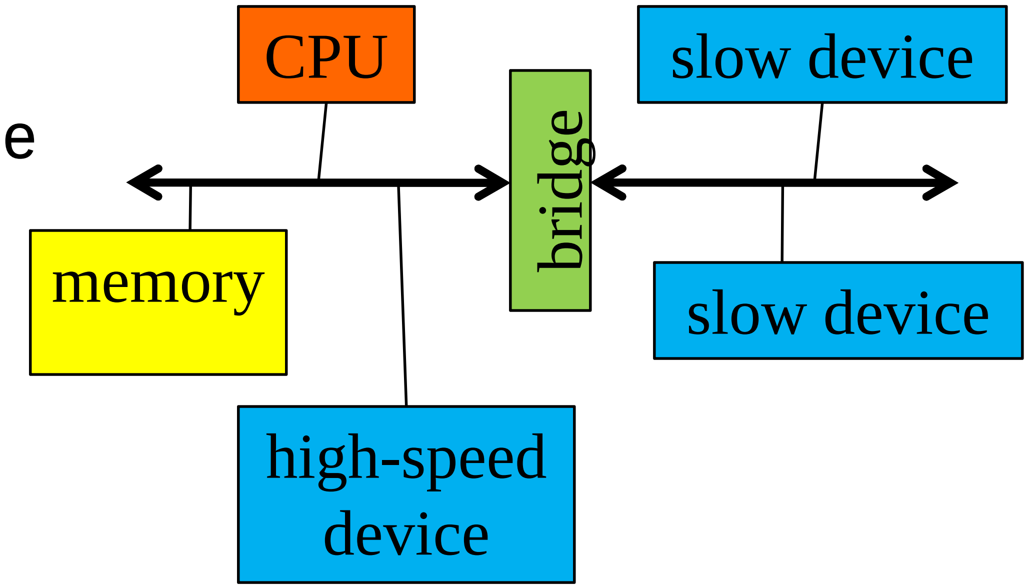


Bus transfer sequence diagram

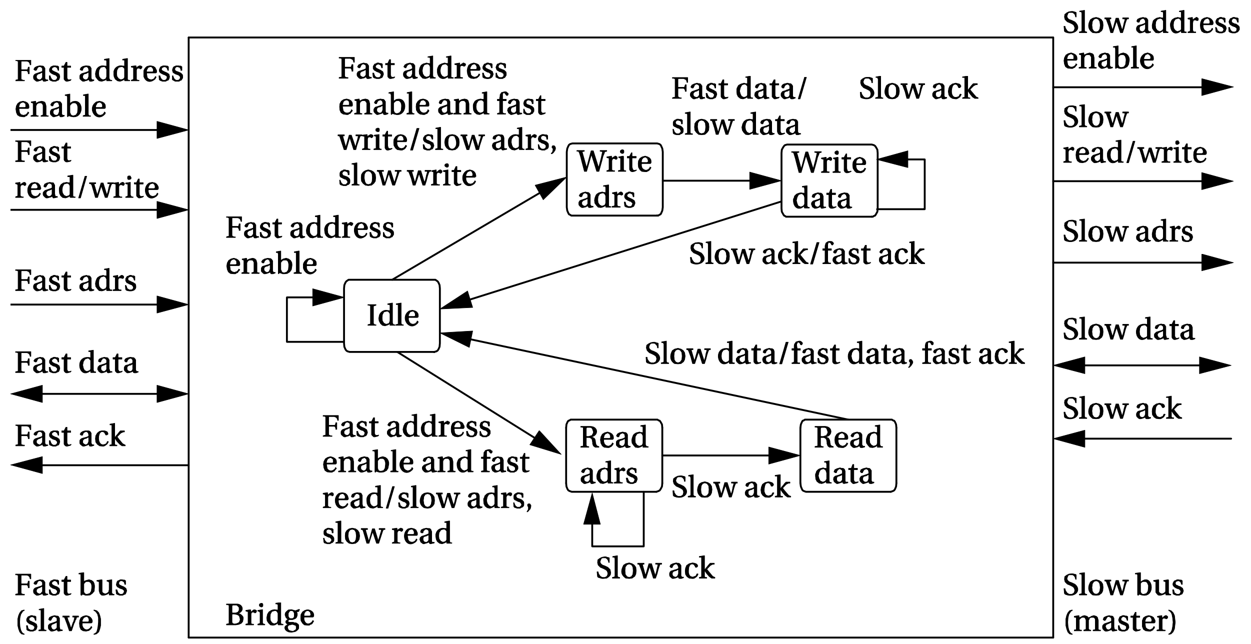


System bus configurations

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

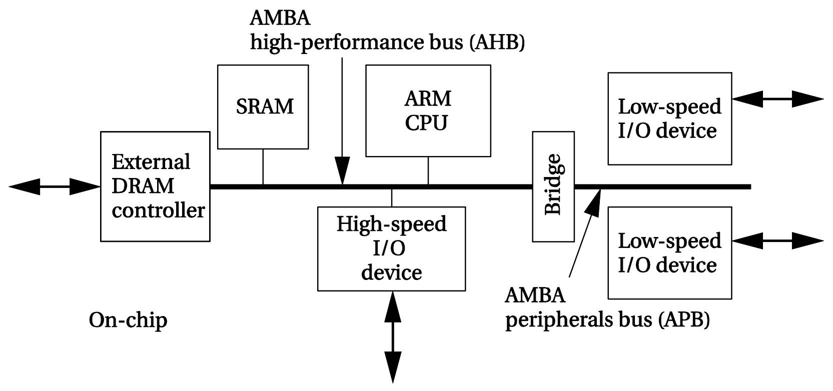


Bridge state diagram



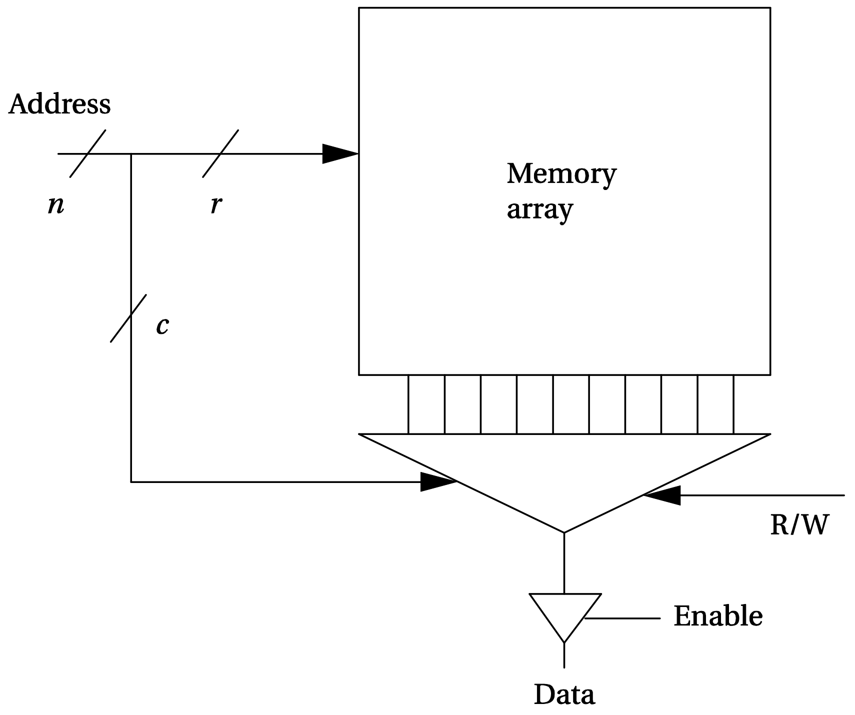
ARM AMBA bus

- 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.



Memory components

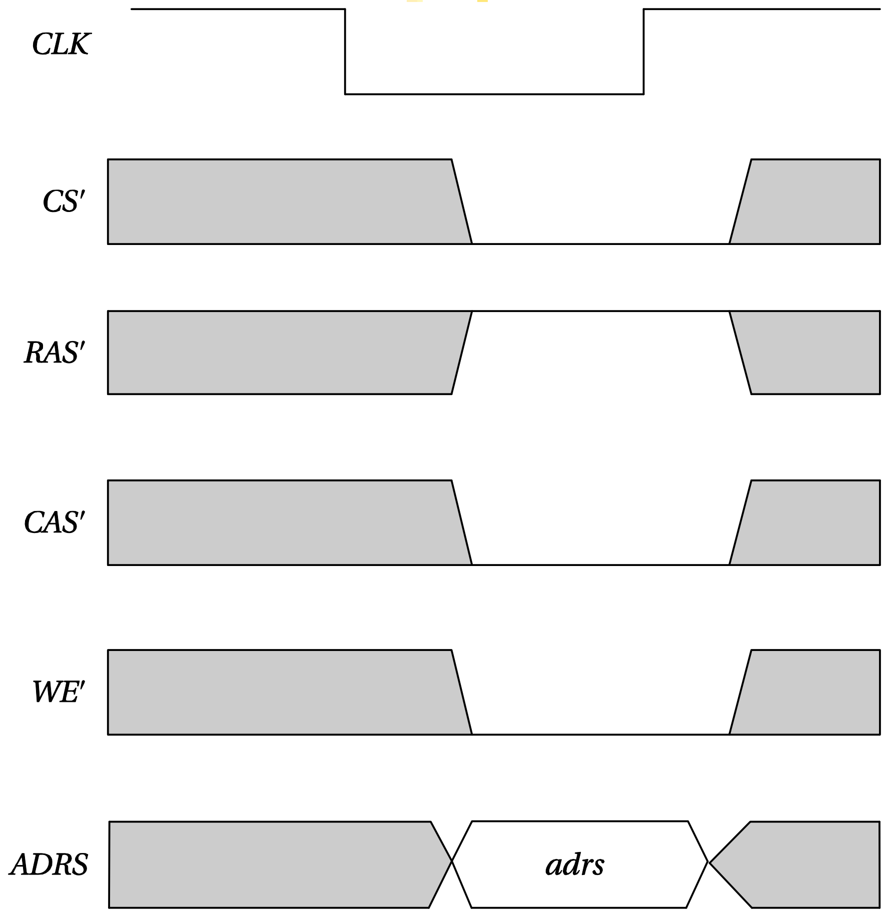
- Several different types of memory:
  - DRAM.
  - SRAM.
  - Flash.
- Each type of memory comes in varying:
  - Capacities.
  - Widths.



Random-access memory

- Dynamic RAM is dense, requires refresh.
  - Synchronous DRAM is dominant type.
  - SDRAM uses clock to improve performance, pipeline memory accesses.
- Static RAM is faster, less dense, consumes more power.

SDRAM operation



Read-only memory

- ROM may be programmed at factory.
- Flash is dominant form of field-programmable ROM.
  - Electrically erasable, must be block erased.
  - Random access, but write/erase is much slower than read.
  - NOR flash is more flexible.
  - NAND flash is more dense.

Flash memory

- Non-volatile memory.
  - Flash can be programmed in-circuit.
- Random access for read.
- To write:
  - Erase a block to 1.
  - Write bits to 0.

Flash writing

- Write is much slower than read.
  - 1.6 μs write, 70 ns read.
- Blocks are large (approx. 1 Mb).
- Writing causes wear that eventually destroys the device.
  - Modern lifetime approx. 1 million writes.

Types of flash

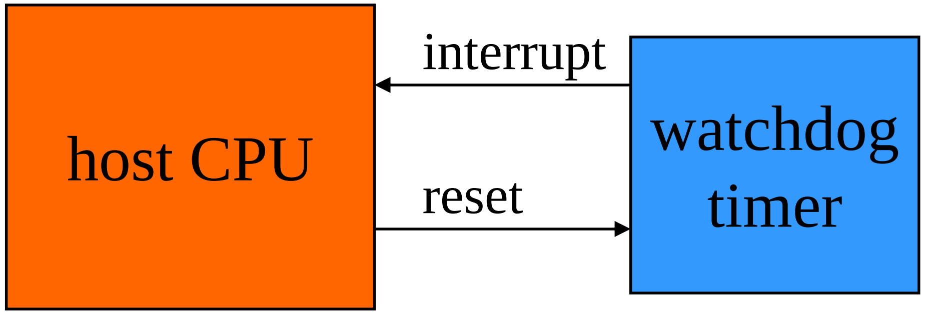
- NOR:
  - Word-accessible read.
  - Erase by blocks.
- NAND:
  - Read by pages (512-4K bytes).
  - Erase by blocks.
- NAND is cheaper, has faster erase, sequential access times.

Timers and counters

- Very similar:
  - a timer is incremented by a periodic signal;
  - a counter is incremented by an asynchronous, occasional signal.
- Rollover causes interrupt.

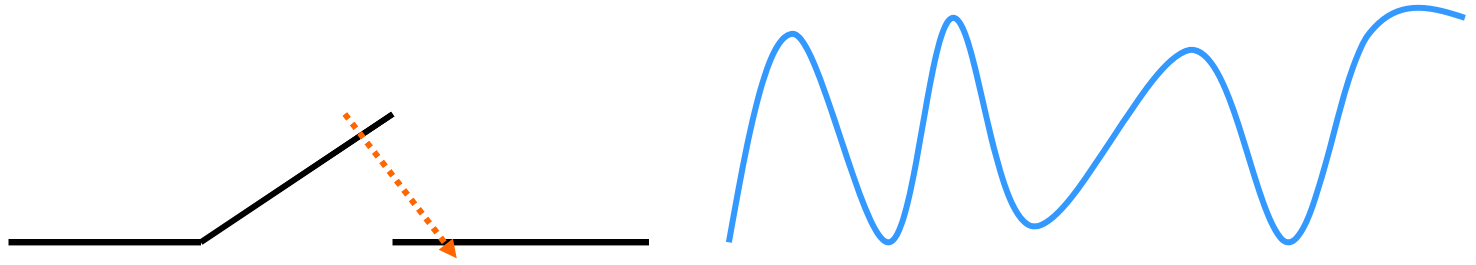
Watchdog timer

- Watchdog timer is periodically reset by system timer.
- If watchdog is not reset, it generates an interrupt to reset the host.



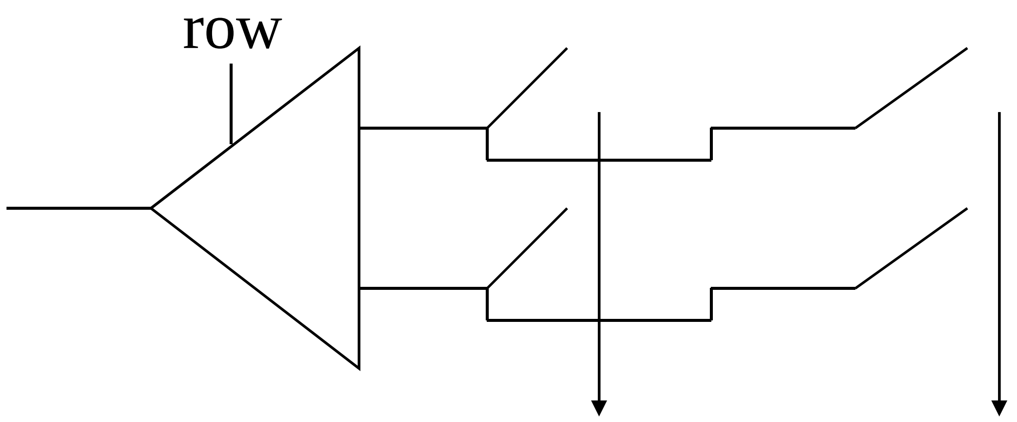
Switch debouncing

- A switch must be debounced to multiple contacts caused by eliminate mechanical bouncing:



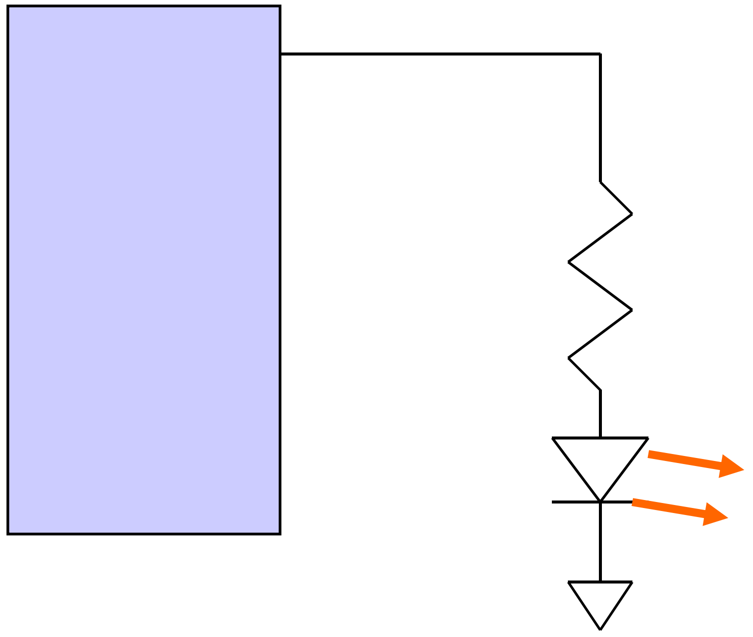
Encoded keyboard

- An array of switches is read by an encoder.
- N-key rollover remembers multiple key depressions.



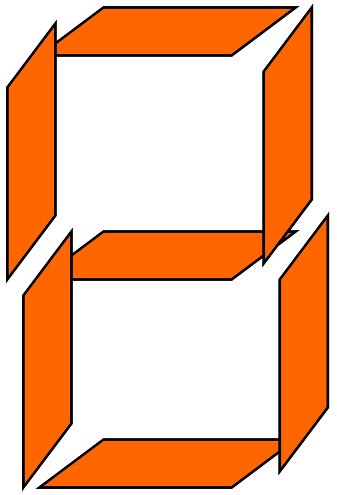
LED

- ⌘Must use resistor to limit current:



7-segment LCD display

- May use parallel or multiplexed input.

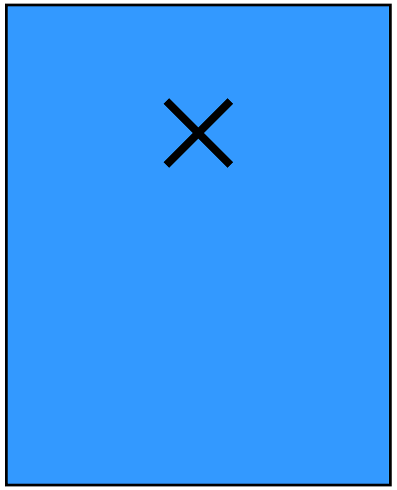


High-resolution display

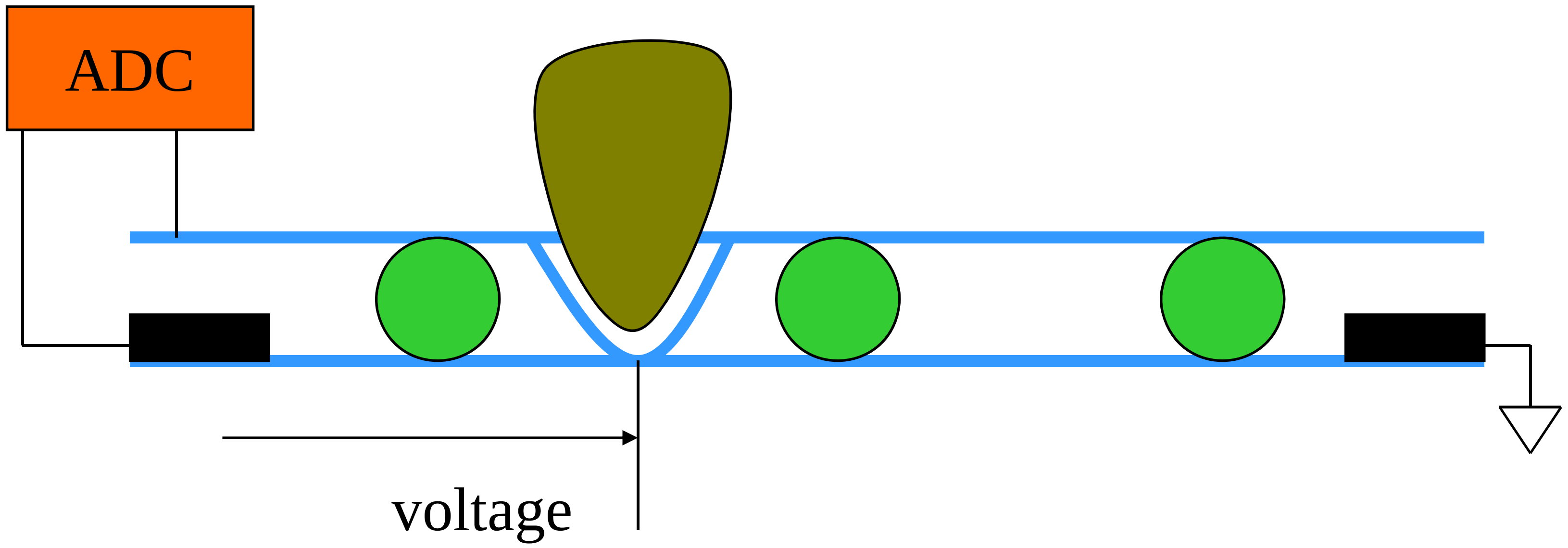
- Liquid crystal display (LCD) is dominant form.
- Plasma, OLED, etc.
- Frame buffer holds current display contents.
  - Written by processor.
  - Read by video.

Touchscreen

- Includes input and output device.
- Input device is a two-dimensional voltmeter:

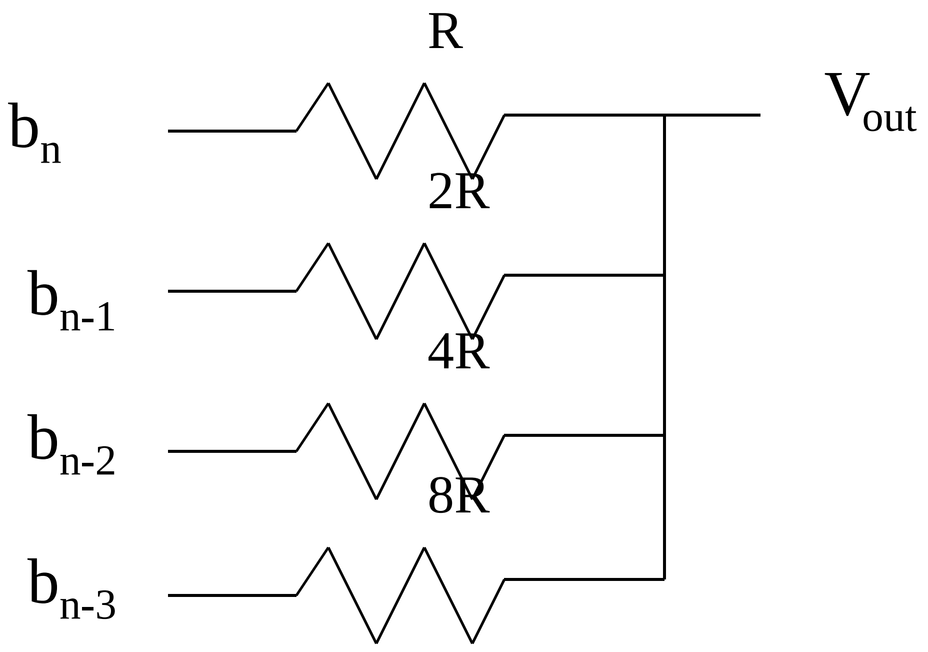


Touchscreen position sensing



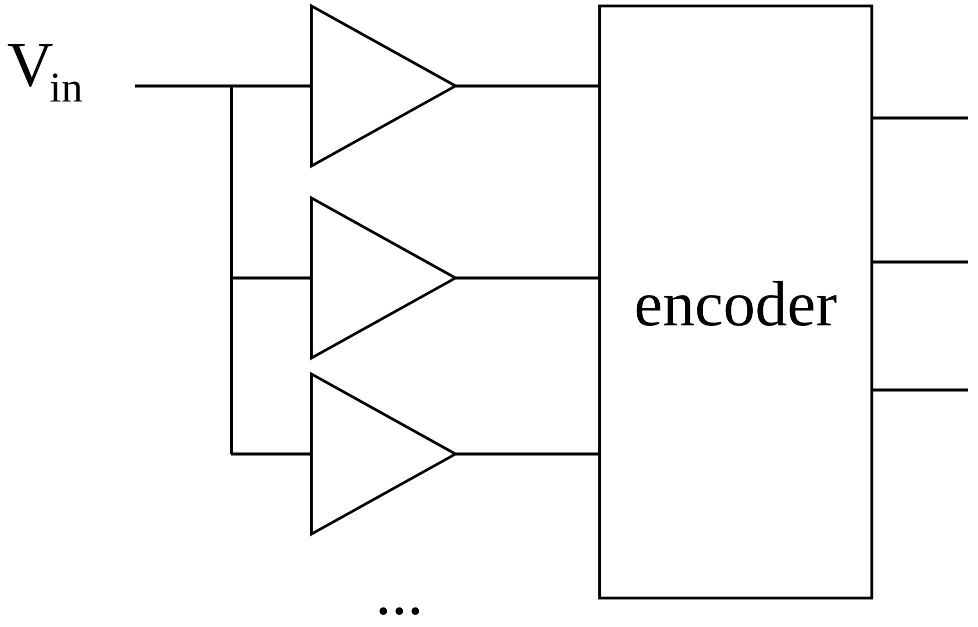
Digital-to-analog conversion

⌘Use resistor tree:



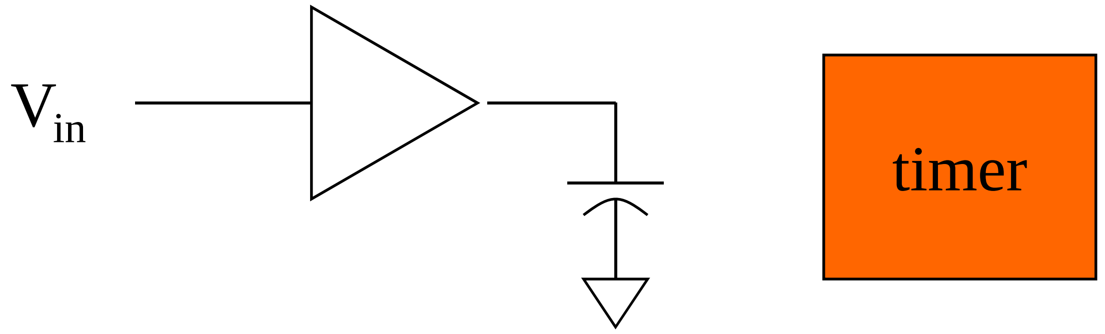
Flash A/D conversion

⌘N-bit result requires 2^n comparators:



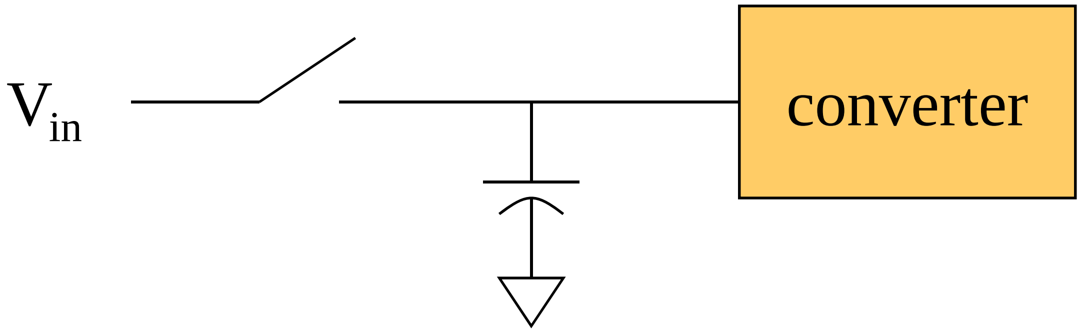
Dual-slope conversion

- Use counter to time required to charge/discharge capacitor.
- Charging, then discharging eliminates non-linearities.



Sample-and-hold

- Samples data:



System architectures

- Architectures and components:
  - software;
  - hardware.
- Some software is very hardware-dependent.