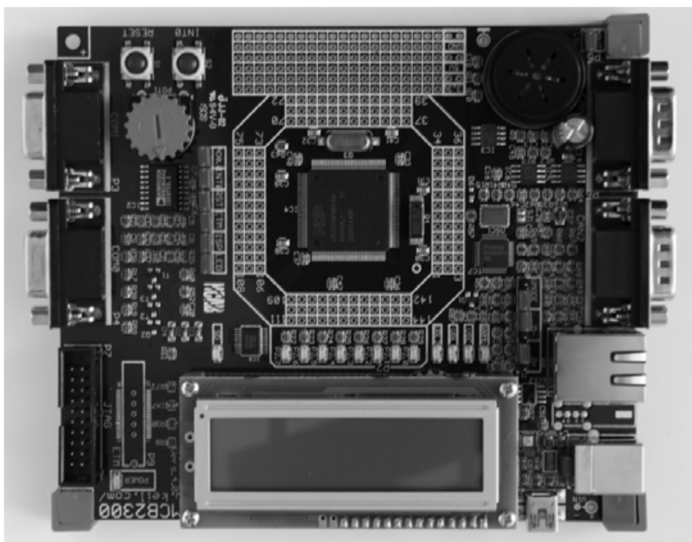
# Computing platforms

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

# Evaluation boards

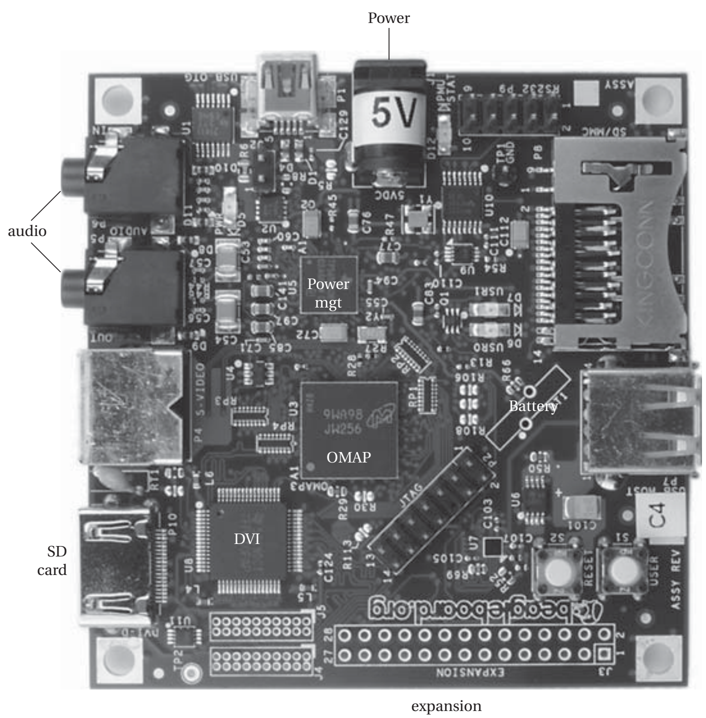
- Designed by CPU manufacturer or others.
- Includes CPU, memory, some I/O devices.
- May include prototyping section.
- CPU manufacturer often gives out reference design---can be used as starting point for your custom board design.

# ARM evaluation module



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

# BeagleBoard




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

# 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

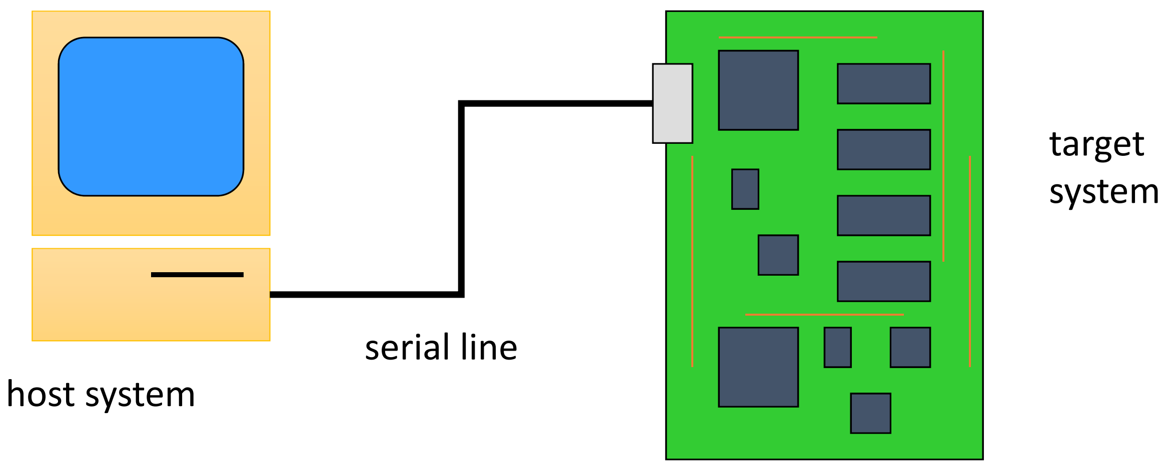
- PCB schematics and artwork files.
- Bill of materials for components.
- Compiler.
- Linux.

# 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

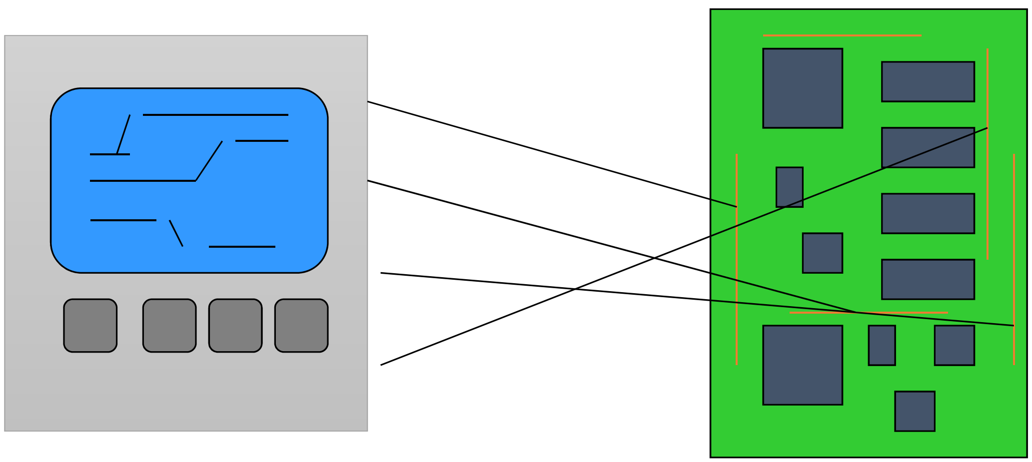
- Save registers.
- Allow user to examine machine.
- Before returning, restore system state.
  - Safest way to execute the instruction is to replace it and execute in place.
  - Put another breakpoint after the replaced breakpoint to allow restoring the original breakpoint.

# In-circuit emulators

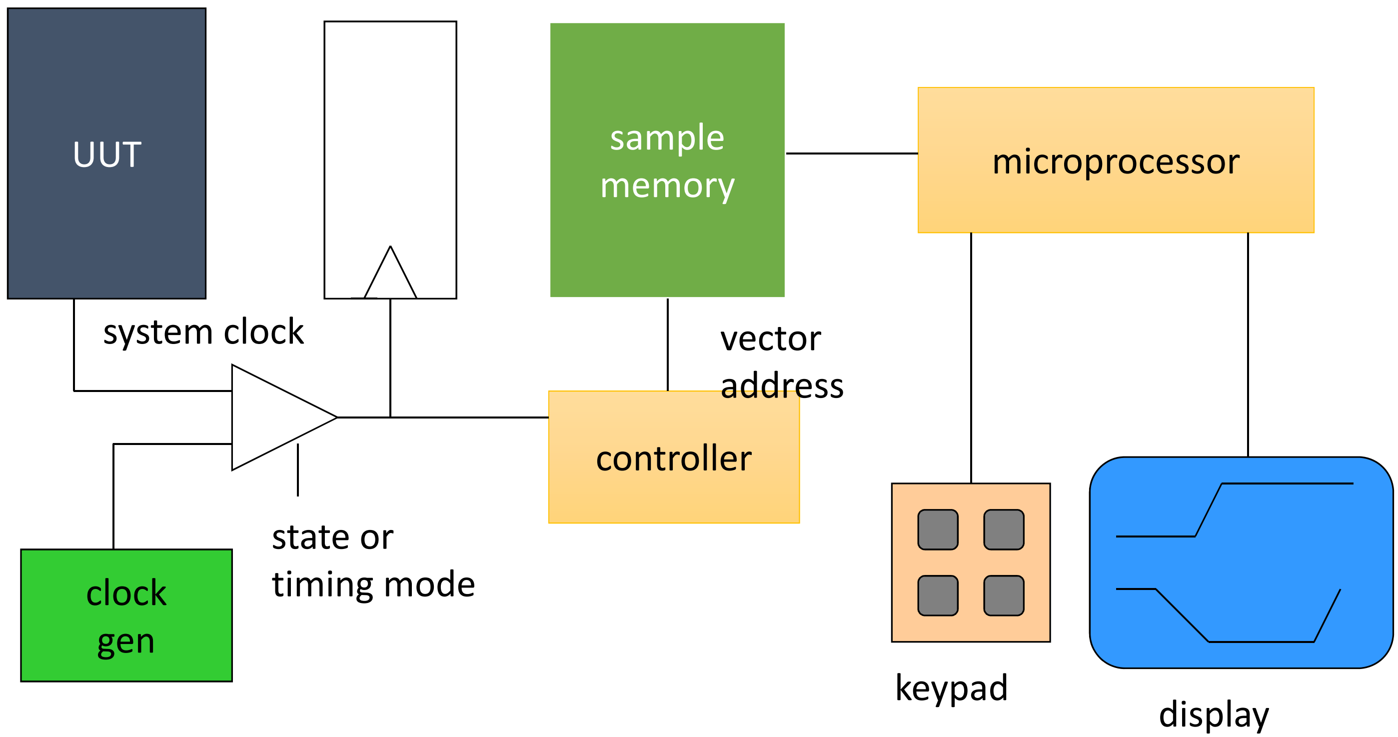
- A microprocessor in-circuit emulator is a specially-instrumented microprocessor.
- Allows you to stop execution, examine CPU state, modify registers.

# Logic analyzers

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

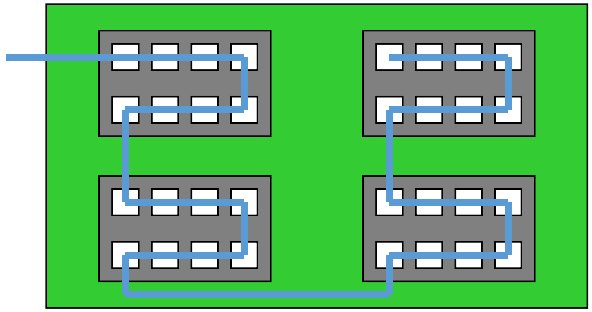


# Logic analyzer architecture



# 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

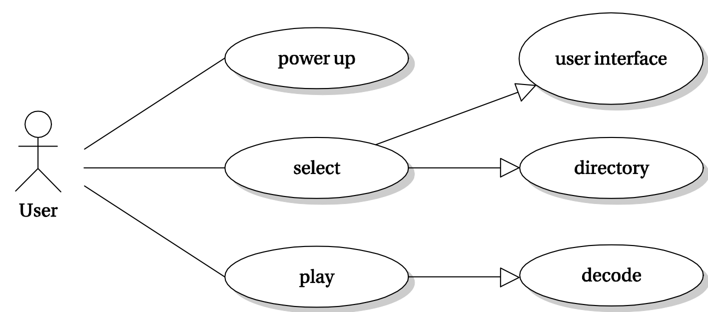
- Run on host system.
- Run on target system.
- Run in instruction-level simulator.
- Run on cycle-accurate simulator.
- Run in hardware/software co-simulation environment.

# Debugging real-time code

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

# 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.

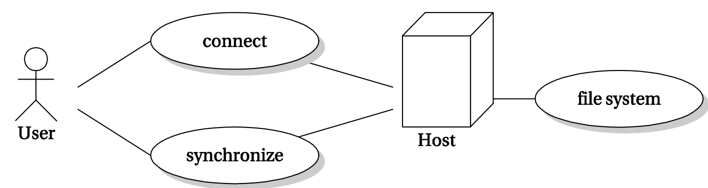


# Non-functional requirements for CE

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

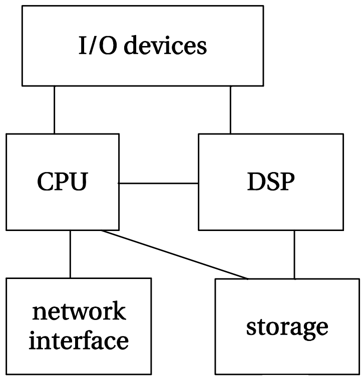
# CE devices and hosts

- 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.

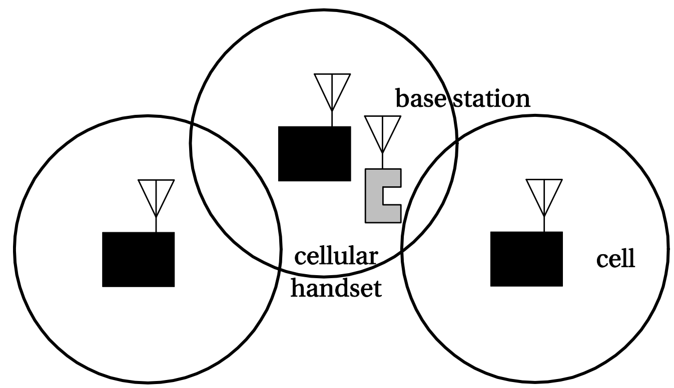


# Flash file systems

- 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.

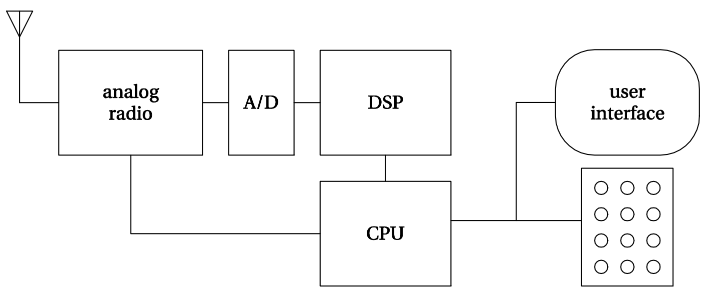
# Cell phones

- 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.



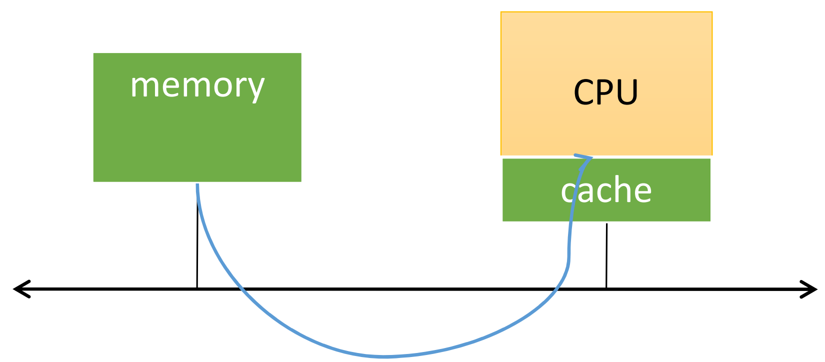
# Cell phone platforms

- 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

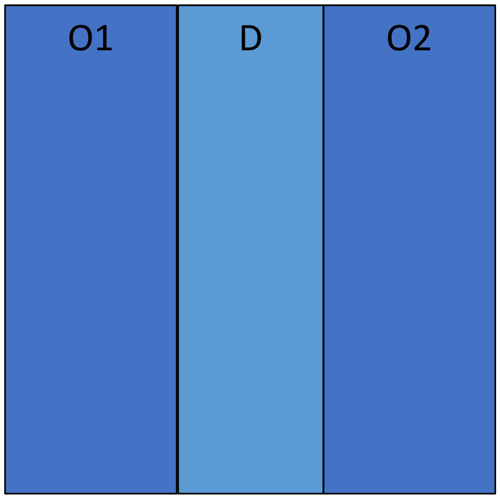
- Video frame: 1920 x 1080 x 3 = 6.2 MBytes.
  - Transfer in 1/30 sec = 0.033 sec,
- 100 MHz 8-bit wide bus requires 0.062 sec/frame---too slow.
- Increase bus width to 32 bits, transfer time reduced to 0.015 sec/frame.
- Increase bus speed to 200 MHz, transfer time is 0.031 frame/sec for8-bit wide bus.

# Bus bandwidth modeling

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

# Bus bandwidth

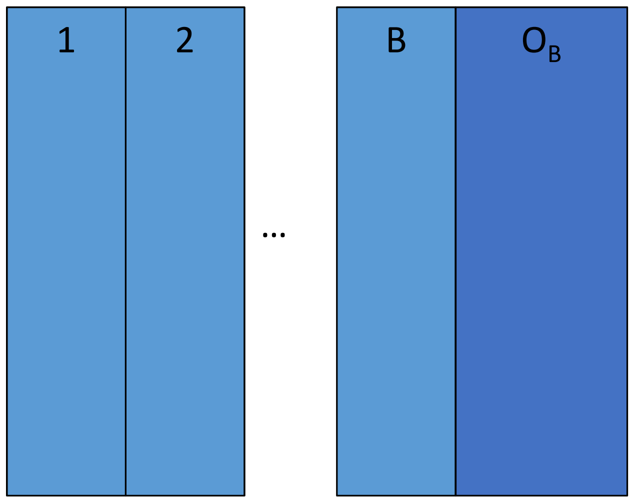
- Assume one word per transfer, compute time for N word transfer.
- Data transfer time D (may include wait states).
- Overhead O1 + O2 = O.




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

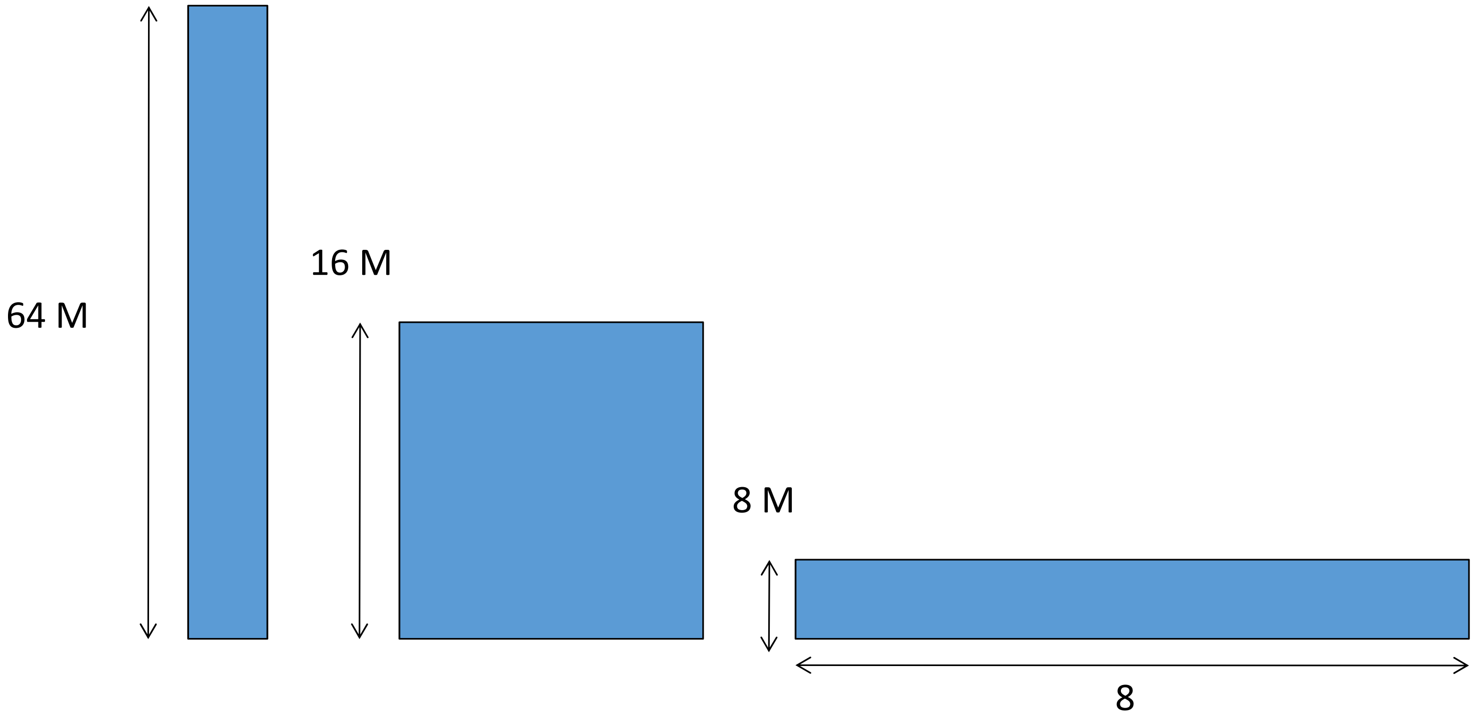
# Bus burst transfer bandwidth

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



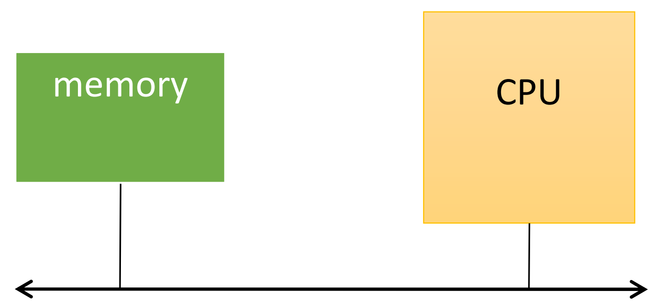

$$T_{burst}(N) = \frac{N}{B}(BD + O_B)$$

# 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.

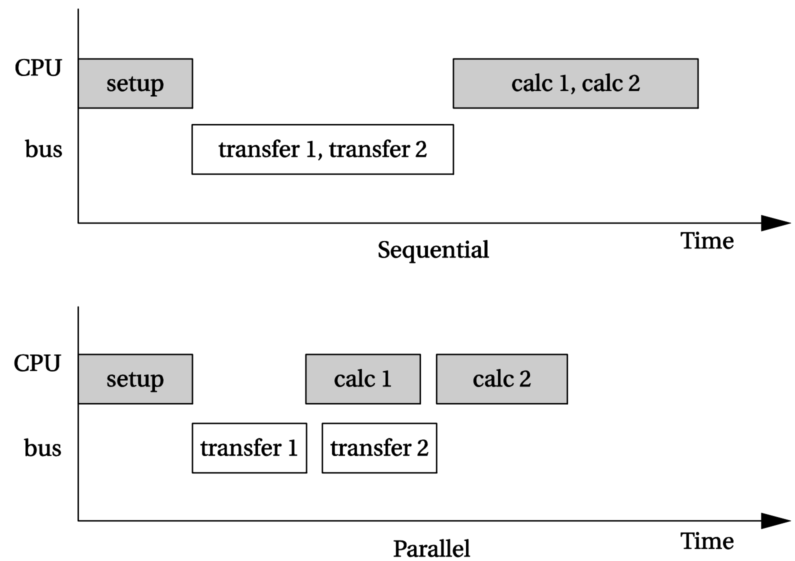
- Bus: assume 100 MHz bus, D=1, O=3:
  - $T_{basic}(1920 \times 1080) = (3+1) \cdot \left(\frac{6.2 \times 10^6}{2}\right) = 12.4 \times 10^6\ cycles$
  - $t_{basic} = T_{basic}P = 0.124\ sec$
- Memory: try burst mode B=4, two bytes wide.
  - $T_{basic}(1920 \times 1080) = \frac{(6.2 \times 10^6/2)}{4}(4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

# 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.