CPU emulators

A quick look on their types, principles of design and operation

Grigory Rechistov

Intel–MIPT lab, MDSP project

October 2010

The definition

Emulator — a program that executes models of certain pieces of hardware.

The definition

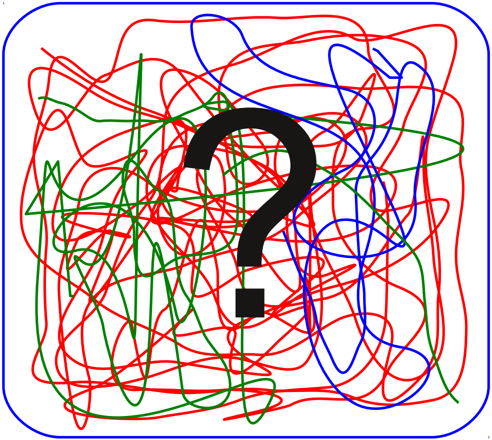
Emulator — a program that executes models of certain pieces of hardware.
Actually, there is no precise definition.

Emulation versus *Simulation*

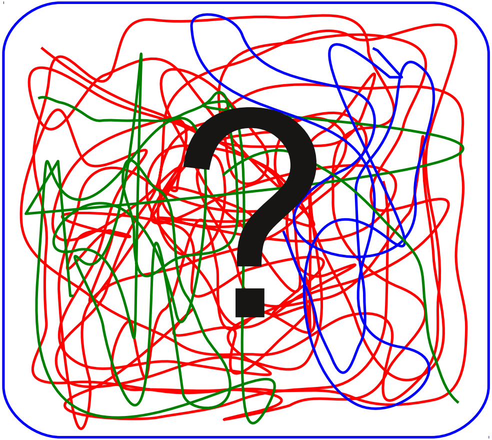
Many meanings exist. I just interchange both terms.
≪Emulation≫ was introduced in 1957 by IBM and since then it was reinterpreted many times.
In Russian, we usually use simple «Моделирование».

How a simulator is usually designed

How a simulator is usually designed



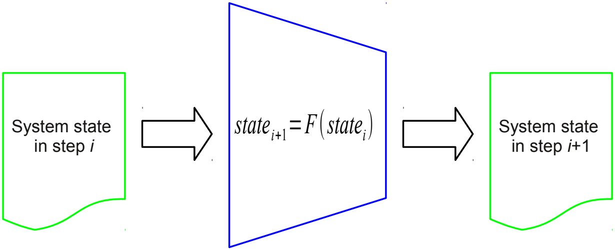
How a simulator is usually designed



. . . But we can build one step by step!

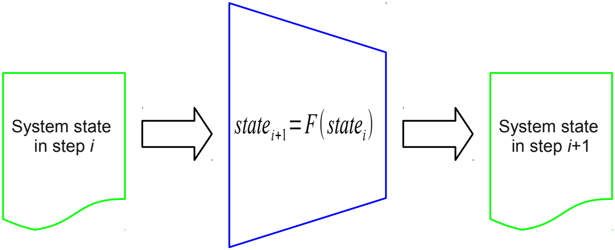
Functional model

Simulates only the functional features of hardware.



Functional model

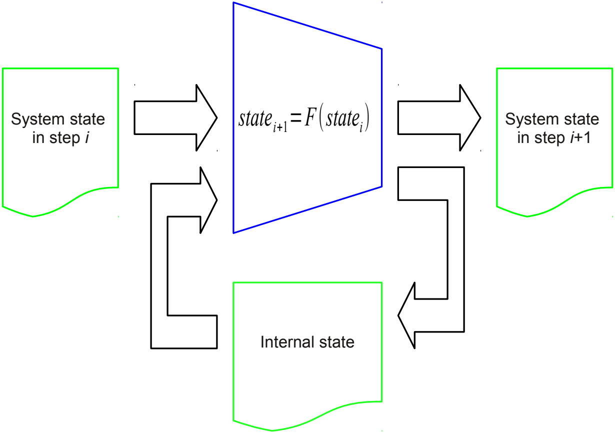
Simulates only the functional features of hardware.



Question: what can be included in system state?

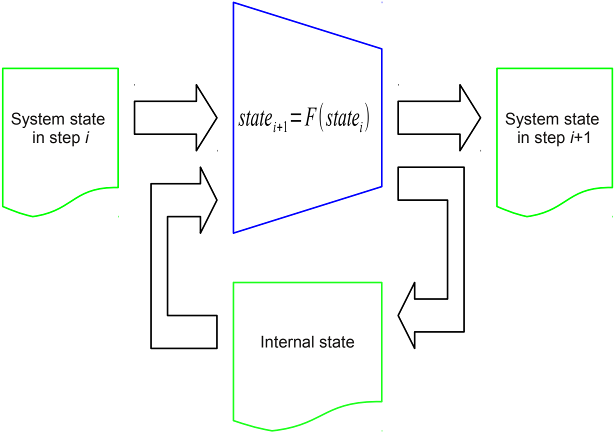
Clock-precise model

Sometimes is called «Performance». Adds a concept of internal temporal state.



Clock-precise model

Sometimes is called «Performance». Adds a concept of internal temporal state.

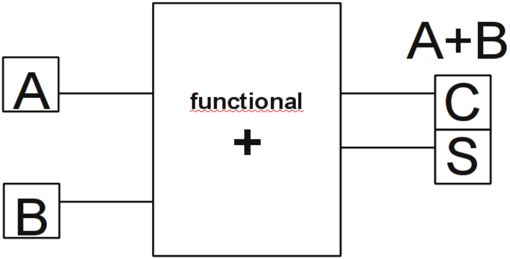


Question: what is in temporal state?

A bit of distraction: a 1 bit summator

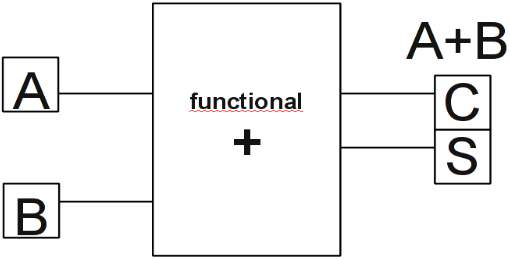
A bit of distraction: a 1 bit summator

Functional model:



A bit of distraction: a 1 bit summator

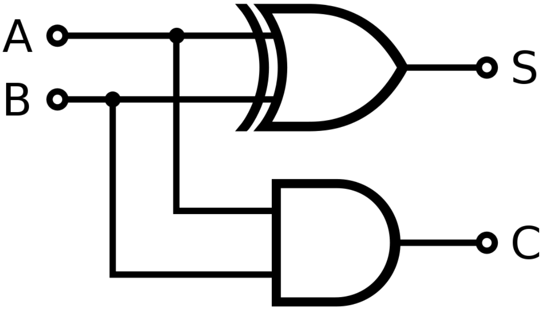
Functional model:



Task: write a table of boolean truth for this adder.

Real 1 bit half-summator

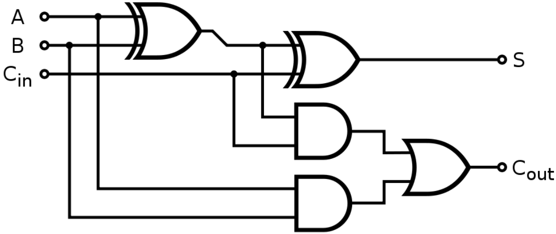
Real 1 bit half-summator



Note: clock synchronization circuitry is not shown.

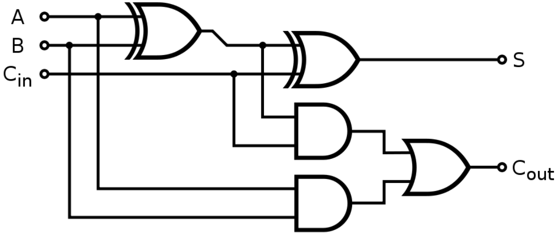
2 bit full summator with latency

2 bit full summator with latency



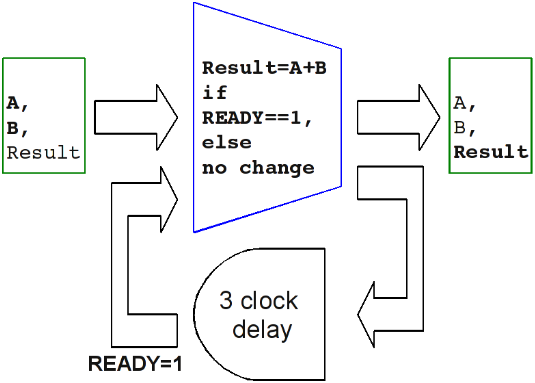
Note: clock synchronization circuitry is not shown.

2 bit full summator with latency

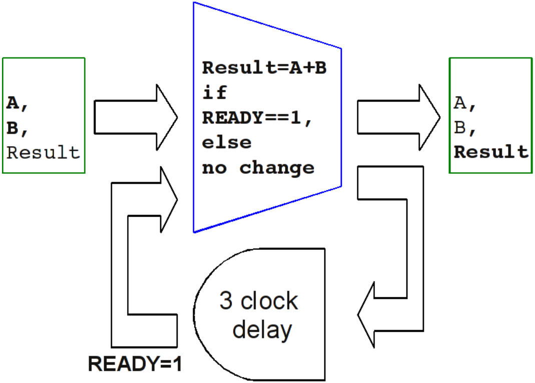


Note: clock synchronization circuitry is not shown.
Question: what type of model this scheme is — functional or clock-precise?

A clock-presise variant of model of full 2 bit adder



A clock-presise variant of model of full 2 bit adder



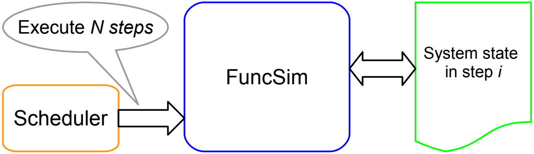
Please note that we don't specify the size of numbers being added or any details of adder. We simply modelled functions **and** timings!

Standalone functional model

Attempts to be on its own. But for this it needs an entity with notion of time!

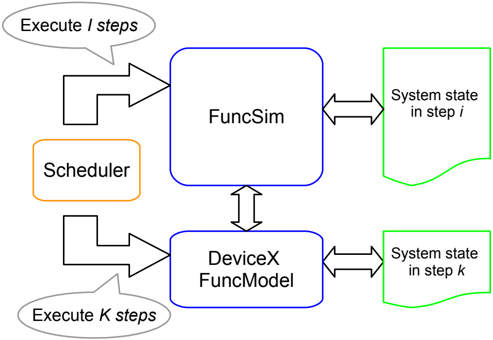
Standalone functional model

Attempts to be on its own. But for this it needs an entity with notion of time!



Full platform functional models

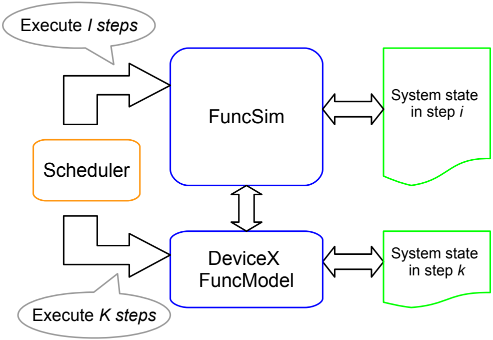
Simulates only the functional features of hardware.



1. Hardly can be used to measure speed, performance, power or certain other parameters of HW.
2. Are able to emulate quite complex systems — from booting OS to airplane complexes.
3. Comparatively fast emulation (slowdown $\times 2 - \times 100$).

Full platform functional models

Simulates only the functional features of hardware.



1. Hardly can be used to measure speed, performance, power or certain other parameters of HW.
2. Are able to emulate quite complex systems — from booting OS to airplane complexes.
3. Comparatively fast emulation (slowdown $\times 2 - \times 100$).

Simulation of time

Problem: there are many devices to simulate and only one time line. . .

Simulation of time

Problem: there are many devices to simulate and only one time line. . .

It is a task of scheduling similar to one performed in an OS.

- Cooperative multitasking
- Preemptive multitasking

Simulation of time

Problem: there are many devices to simulate and only one time line. . .
It is a task of scheduling similar to one performed in an OS.

- Cooperative multitasking
- Preemptive multitasking

Another issue is

- Problem of synchronization of all devices' point of view on time.

Types of emulators one more time.

Application mode provide just minimal set of components able to run the particular workloads, e.g. CPU, memory, IO.
Full platform simulates the complete set of HW found in a particular computing system: CPU, memory, network, sound, display, disk, keyboard/mouse. . .
Hybrid use a junction of software components with ones modeled with hardware models e.g. on FPGA.
Distributed are placed on several computers and interact over the network.

Application mode

- Usually the first one to be implemented for a new architecture.
- Cannot simulate any OS booting (as if there is any OS for the *new* architecture!)
- But applications need an OS to work! Thus such simulator has to implement some minimal non-architectural ABI.
- Can be used to build some performance simulator and let it measure speed etc [6].

Application mode

- Usually the first one to be implemented for a new architecture.
- Cannot simulate any OS booting (as if there is any OS for the *new* architecture!)
- But applications need an OS to work! Thus such simulator has to implement some minimal non-architectural ABI.
- Can be used to build some performance simulator and let it measure speed etc [6].

Application mode

- Usually the first one to be implemented for a new architecture.
- Cannot simulate any OS booting (as if there is any OS for the *new* architecture!)
- But applications need an OS to work! Thus such simulator has to implement some minimal non-architectural ABI.
- Can be used to build some performance simulator and let it measure speed etc [6].

Application mode

- Usually the first one to be implemented for a new architecture.
- Cannot simulate any OS booting (as if there is any OS for the *new* architecture!)
- But applications need an OS to work! Thus such simulator has to implement some minimal non-architectural ABI.
- Can be used to build some performance simulator and let it measure speed etc [6].

Application mode

- Usually the first one to be implemented for a new architecture.
- Cannot simulate any OS booting (as if there is any OS for the *new* architecture!)
- But applications need an OS to work! Thus such simulator has to implement some minimal non-architectural ABI.
- Can be used to build some performance simulator and let it measure speed etc [6].

General steps in making an emulator [4]

- Model the CPU and memory.
- Emulate an instruction set, create disassembler.
- Stub out the rest of the architecture.
- Get basic IO working.
- Work on virtualizing the remaining hardware.

The simpliest CPU emulation [3]

```
for(;;) {
    OpCode = Memory[PC];
    PC++; // program counter
    Counter -= Cycles[OpCode];
    switch(OpCode) {
        case OpCode1:
            Simulate1(); break;
        case OpCode2:
            Simulate2(); break;
        //...
    }
    if(Counter <= 0) {
        // check for interrupts and other tasks
        Counter += InterruptPeriod;
        if (ExitRequired) break;
    }
}
```

Memory emulation

The simplest way to access emulated memory is to treat it as a plain array of items:

```
Data = Memory[Address1];
Memory[Address2] = Data;
```

Such simple memory access is not always possible for following reasons:

- Paged Memory.
- Mirrored Memory.
- ROM protection.
- Memory-Mapped I/O. Accesses to such memory locations produce ≪special effects≫ and therefore should be tracked.

```
Data=ReadMemory(Address1);
WriteMemory(Address2,Data);
```

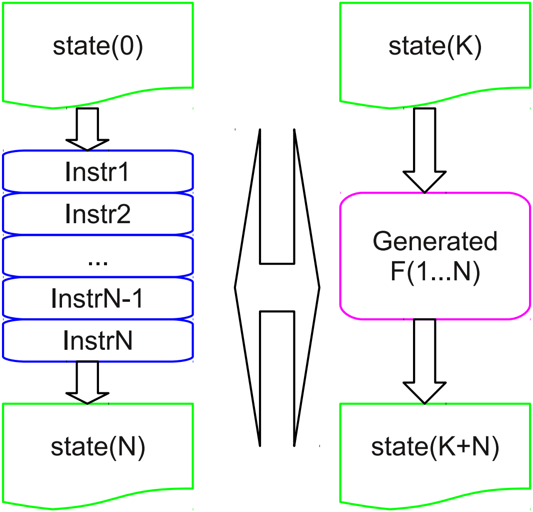
A bit low on speed?

The common rule: don't do the work already done.

A bit low on speed?

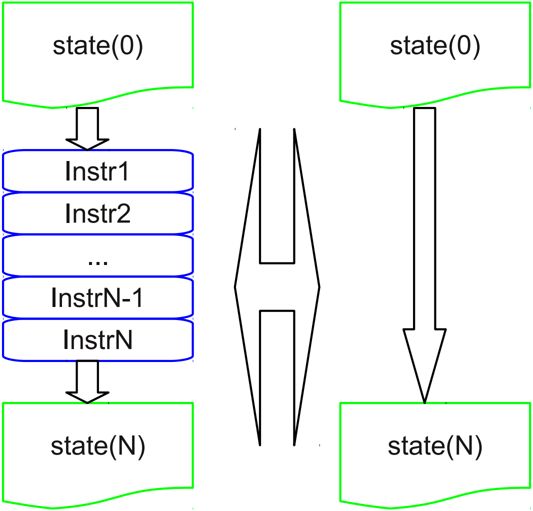
The common rule: don't do the work already done.

Traces



A bit low on speed?

Hyper Simulation



How to boost the speed

Wide range of techniques is used

- Not so lame interpretation: hashing, lookup tables.
- Binary translation.
- Just-in-time compilation.
- Direct execution using virtualization.

Quiz

1. When was term ≪emulation≫ invented?

Quiz

1. When was term ≪emulation≫ invented? In 1957.

Quiz

1. When was term ≪emulation≫ invented? In 1957.
2. What type of emulators is the slowest?

Quiz

1. When was term ≪emulation≫ invented? In 1957.
2. What type of emulators is the slowest? Performance

Quiz

1. When was term ≪emulation≫ invented? In 1957.
2. What type of emulators is the slowest? Performance
3. What type of emulators is the most accurate?

Quiz

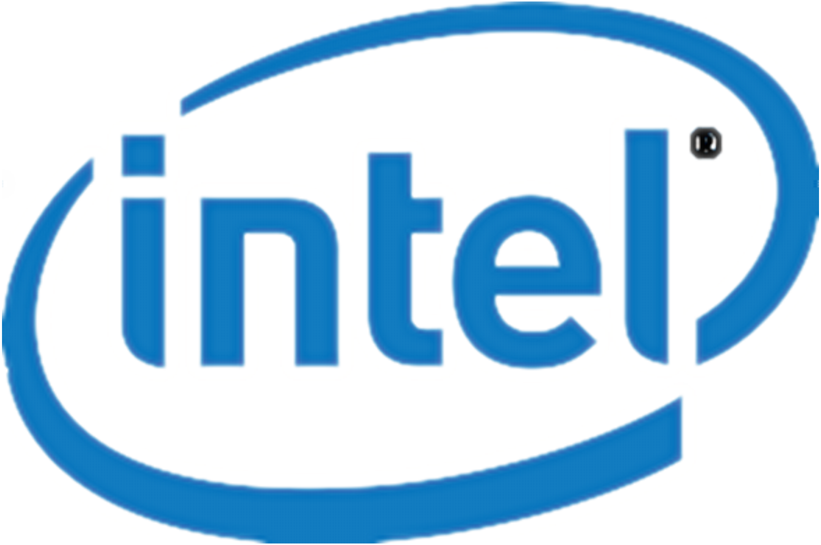
1. When was term ≪emulation≫ invented? In 1957.
2. What type of emulators is the slowest? Performance
3. What type of emulators is the most accurate? it depends on task.

Quiz

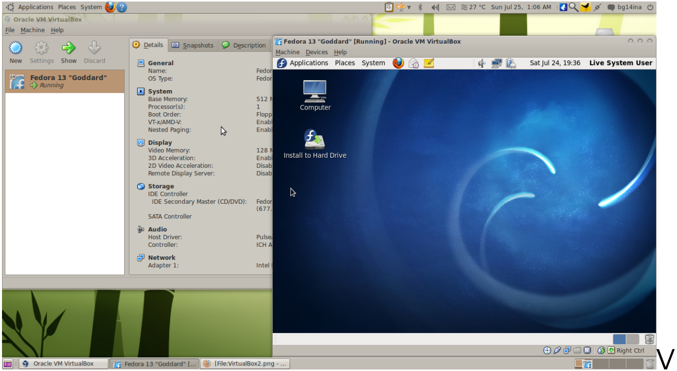
1. When was term ≪emulation≫ invented? In 1957.
2. What type of emulators is the slowest? Performance
3. What type of emulators is the most accurate? it depends on task.

More to read

- Крис Касперски. Техника оптимизации программ. Эффективное использование памяти. СПб. БХВ-Петербург, 2003.
- Wikipedia `http://en.wikipedia.org/wiki/Emulator`
- Marat Fayzullin. How To Write a Computer Emulator `http://fms.komkon.org/EMUL8/HOWTO.html`
- Tony Gray. How to Write an Emulator `http://www.tucs.org.au/how-to-write-an-emulator/`
- Daniel Boris. How Do I Write an Emulator?, Part 1, R1.00
- Carole Dulong et al. The Making of a Compiler for the Intel Itanium Processor. Intel Technology Journal Q3, 2001


intel®

Emulators


Applications Places System
27 °C Sun Jul 25, 1:06 AM
bg14ina
Oracle VM VirtualBox
File Machine Help
New Settings Show Discard
Fedora 13 "Goddard"
Running
Details Snapshots Description
Fedora 13 "Goddard" [Running] - Oracle VM VirtualBox
Machine Devices Help
Applications Places System
Sat Jul 24, 19:36 Live System User
Computer
Install to Hard Drive
Right Ctrl
Oracle VM VirtualBox
Fedora 13 "Goddard" [...
[File:VirtualBox2.png - ...

General
Name: Fedor
OS Type: Fedor

System
Base Memory: 512 M
Processor(s): 1
Boot Order: Flopp
VT-x/AMD-V: Enabl
Nested Paging: Enabl

Display
Video Memory: 128 M
3D Acceleration: Enabl
2D Video Acceleration: Disab
Remote Display Server: Disab

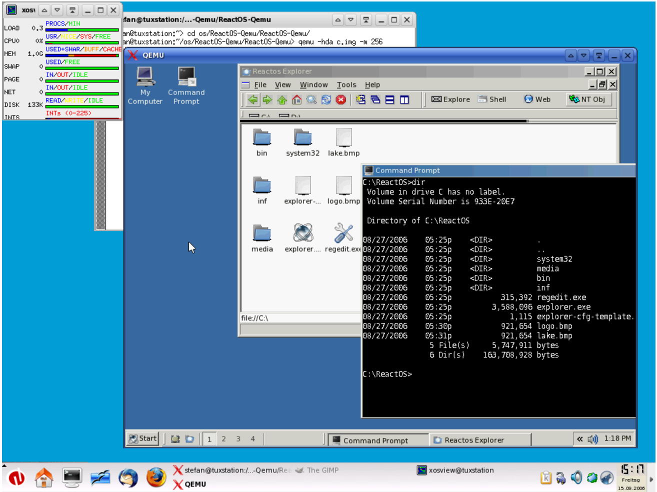
Storage
IDE Controller
IDE Secondary Master (CD/DVD): Fedor (677.
SATA Controller

Audio
Host Driver: Pulse
Controller: ICH A

Network
Adapter 1: Intel

VirtualBox running Fedora 13.

Emulators

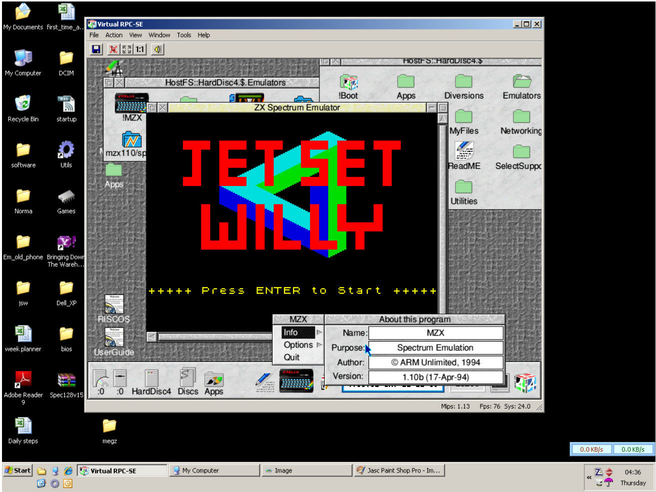


```
an@tuxstation:~> cd os/ReactOS-Qemu/ReactOS-Qemu/
an@tuxstation:~/os/ReactOS-Qemu/ReactOS-Qemu> qemu -hda c.img -m 256
```

```
C:\ReactOS>dir
 Volume in drive C has no label.
 Volume Serial Number is 933E-20E7

 Directory of C:\ReactOS

08/27/2006   05:25p    <DIR>          .
08/27/2006   05:25p    <DIR>          ..
08/27/2006   05:25p    <DIR>          system32
08/27/2006   05:25p    <DIR>          media
08/27/2006   05:25p    <DIR>          bin
08/27/2006   05:25p    <DIR>          inf
08/27/2006   05:25p           315,392 regedit.exe
08/27/2006   05:25p         3,588,096 explorer.exe
08/27/2006   05:25p             1,115 explorer-cfg-template.
08/27/2006   05:30p           921,654 logo.bmp
08/27/2006   05:31p           921,654 lake.bmp
               5 File(s)      5,747,911 bytes
               6 Dir(s)     163,708,928 bytes

C:\ReactOS>
```

Qemu running ReactOS.

Emulators



About this program

Name: MZX
Purpose: Spectrum Emulation
Author: © ARM Unlimited, 1994
Version: 1.10b (17-Apr-94)

Virtual RPC running MZX running ZX Spectrum.

Emulators, lots of them! [2]

Emulators, lots of them! [2]

- Oracle VirtualBox

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta
- AMD SimNOW!

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta
- AMD SimNOW!
- WindRiver Simics

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta
- AMD SimNOW!
- WindRiver Simics
- Parallels

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta
- AMD SimNOW!
- WindRiver Simics
- Parallels
- VMWare Workstation and Server

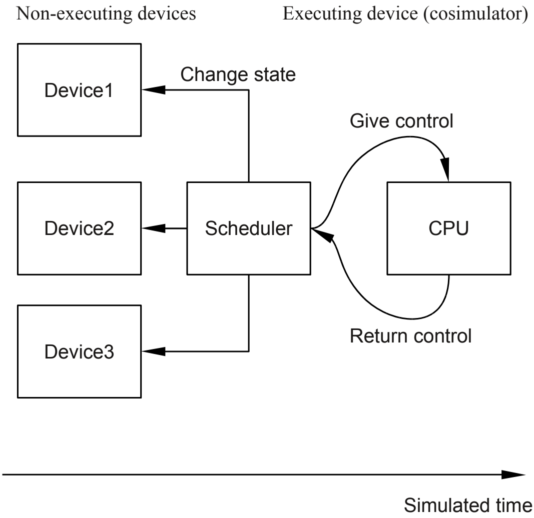
Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta
- AMD SimNOW!
- WindRiver Simics
- Parallels
- VMWare Workstation and Server
- ARMware

Emulators, lots of them! [2]

- Oracle VirtualBox
- Microsoft VirtualPC
- Bochs
- Qemu
- Apple Rosetta
- AMD SimNOW!
- WindRiver Simics
- Parallels
- VMWare Workstation and Server
- ARMware
- . . . thousands of them! For every and each of architecture including PDP-10, ZX Spectrum, NES and Itanium.

Simulation of time



Other pecularities

- Simulation of MP systems.
- Endiannes.
- Speed of emulation.