Meet us Saturday At 11:00 !

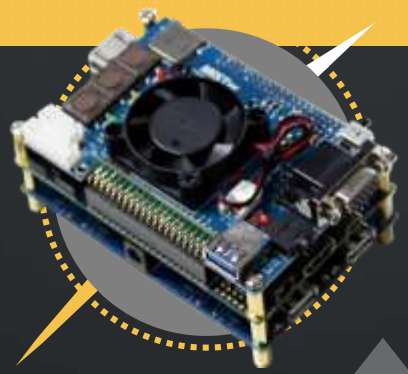
silicium

# Retrocomputer Preservation

**Preserving is not just about storing computers, disks and books in a hangar**

## FOSDEM Brussels, 2 & 3 February 2019


**Original machine ?**



**Hard Emulation ?**

**Soft Emulation ?**


Retrogaming & Retrocomputing

A French non-profit organization for the preservation of videogame and computer heritage

V 2019.02.01

# *our vision

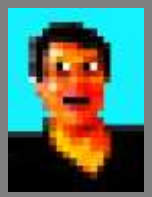
*To preserve heritage videogames and computers. To show them to people. Silicium manages an ever-growing collection and a regular exhibition schedule.*

# about silicium

## A 25-year history

Back in 1994, it was so obvious that computers should be preserved. We then decided to gather and shelter many obsolete computers, videogames, software, books... and testimonies.
Once we were seen like Zorg planet inhabitants, but today everyone feels the need to preserve computer heritage.
Silicium has some answers.

# our team today



## René Speranza

***President & cofounder***

Has never achieved to transcend the 16-color palette.

## Benoit Triquet

***Digital specialist***

Not everyone can whisper the true language of the machine to the computer.

## Valérie Amiel

***Management expert***

Explains us how to pop around things we need.

# *our collection

*The heart of Silicium lies in its unmatched collection of 1965 to 2015 computing representatives, which are readily available for studies, display & exhibitions*

# * our action

It's hard to describe what Silicium preserves.
Here's what we consider important.

## Computers

It all begun with compact Hewlett-Packard HP-85. Instead of being recycled, it launched the idea of computer preservation in 1989.

## Videogames


People ask us to show some fun. Videogames seem to be the right answer. And we know that even the hardcore computer can be fun to play with.

## Everything

Tons of iron, loads of paper. We try to maintain the whole thing together in order to keep the spirit of computing vivid and unforgettable.


# *our experience

*With more than 100 members and a 25-year history, Silicium tries to offer a true savoir-faire.*

# * our expertise

Silicium not only manages a huge collection.
We also promote this heritage in many ways.

## Exhibitions

The main link with people

## Expertise

Need an expert in DEC minicomputers or in Philips Videopac ? Ask Silicium.

## Publisher

Silicium regularly publishes books and games.

## Manage community

Mysilicium forum is a hub for enthusiasts to share their passion.

## Conference

This heritage has a long story to tell.

## Preservation

Unfortunately, many old machines need to be repaired.
Silicium experts play a major role.

# * Silicium in pictures

Some fine examples of Silicium activities.


2013 Odyssud Gameplay was a big exhibition,


Nothing could be done without enthusiasts and a huge collection,


French TV captures an old computer screen drawing live,


Old technology requires some specific skills to maintain machines in working order.

Let’s achieve our main goal : **opening a permanent exhibition.**

# Retrocomputer Preservation

**Preserving is not just about storing computers, disks and books in a hangar**

*We do repair computers but nothing is forever*

**Hardware rots**

Early ASICs in 8-bit British micros…

**Media rots**

Mass-storage emulators

**Documentation rots**

Digitize
Reverse-engineer

# Projects

**Projects**
we use or make
at Silicium

**Retro** Peripherals
**Reverse** Engineering
**FPGA-Based** emulators

**Projects**
by our members

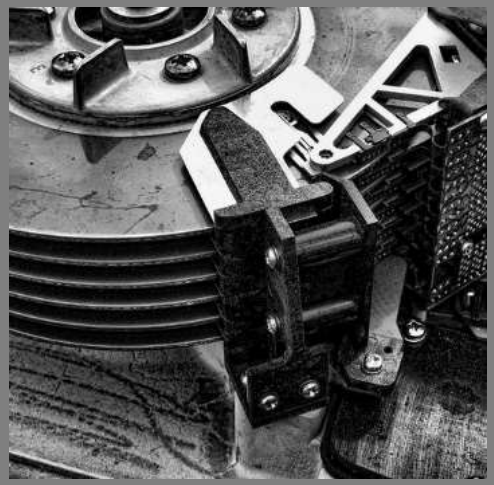
**Gregory Estrade** (@Torlus, http://lvt.tl/)

**Frédéric Requin** (co-founder of ArcadeRetroGaming)

# Techniques for retro development

# * Retro mass-storage devices


Tape Decks



Hard Drives

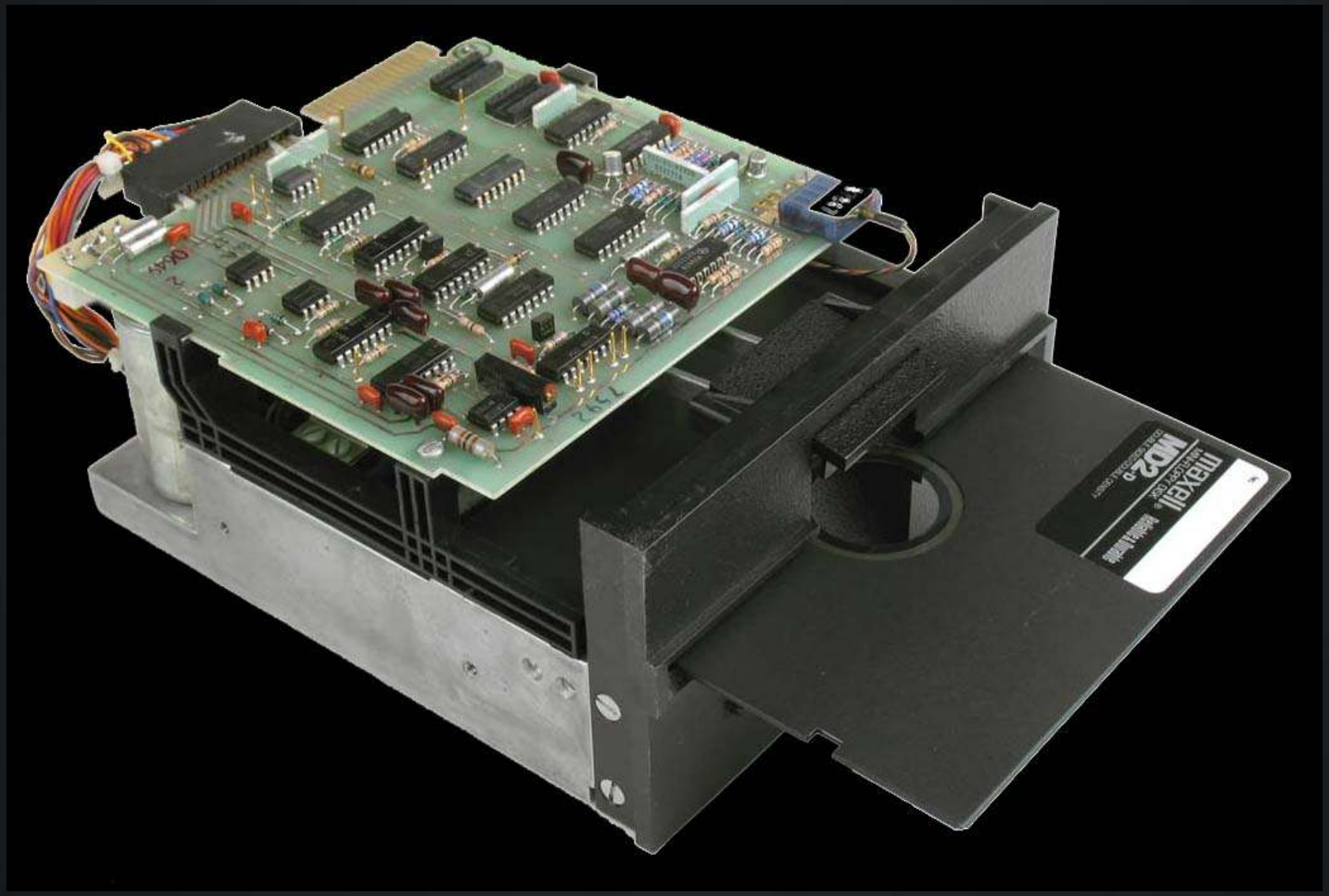
**Audio interface is easy:**
smartphone with headphone jack
Digital interfaces are proprietary
Atari 800: no emulators but A8CAS
open source software for digitizing
copy-protected tapes and
use in emulators

3€ generic « SD2IDE »
work best in our experience
CF-to-IDE are passive, depend on your CF card
IDE-to-mSATA are less compatible
Open source SCSI2SD
We use v5 manufactured by Itead

# Floppy disks

- **A floppy drive is « dumb », mostly analog electronics**
  - Most 8-bit micros do not have a floppy controller
- **Easy: digital serial port**
  - Atari 800: an Arduino is enough, open-source SIO2SD ebay, lotharek.pl
  - BBC Micro: software-defined so anything works, we use Ctorwy31 MMC drive
- **Hard: processor bus**
  - Oric floppy controller boards are remade
- **Standard non-PC controller**
  - Like Apple II where we use CFFA3000, BigMessOfWires FloppyEmu, …

# Shugart-like floppy drives



# Shugart-like floppy drives

- **Beware of pinout differences between true Shugart and PC industry standard drives**
  - Motor-on and drive select lines
- **Non-standard ribbon cables: Amstrad CPC**
  - 26-pin ribbon cable to 34-pin PC => 2-way success
  - Beware of reversed +5V and +12V !!!
- **HxC is the best: resilient hardware, highly compatible**
  - By JF Del Nero with contribs from Torlus
  - Hardware is open source :) firmware is not :(
- **Gotek drives are inexpensive**
  - Must-have: FlashFloppy open source firmware

# * reverse engineering

*The case of the Squale by Apollo 7*

*French microcomputer*
*only 3 prototypes built*
*1 preserved by French museum CNAM*
*Reverse engineered by Greg Estrade*
*and JF Del Nero*
*Emulator now available*

http://hxc2001.free.fr/Squale/
http://torlus.github.io/2015/04/10/cnam-squale

# * FPGA-based emulators

## * Greg

(@Torlus, http://lvt.tl/)

Sega Megadrive/Genesis
Nec PC-Engine/TurboGrafx
Atari Jaguar
all developed in Verilog on Altera Cyclone
Ported to the MiSTer open source project
Successor to the MiST FPGA open source project
Uses Cyclone V eval board, inexpensive, huge FPGA

## * Fred

(co-founder of ArcadeRetroGaming)

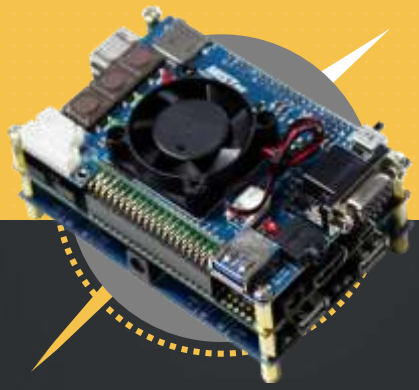
MCC (Multiple Classic Computer) emulates a C64
J68, a Verilog implementation of Motorola 68000

# It's demo time !

## Bomberman '93 running on a...


### PC Engine

A real Nec PC-Engine Is priceless :-) and in excess of 200€ on eBay



### MiSTer

Greg's FPGA core for MiSTer on DE10-nano 140€ to 200€ depending on daughterboards

### Raspberry Pi

Batocera Linux on Raspberry Pi 3B+ 50€

# * Retro hardware development

## Greg and Fred use Verilator

- **Veritalor** open source, Verilog simulator
- **GTKWave** VCD viewer

- **Verification** of J68 core against Musashi, the 68k emulator used by MAME
- **Cosimulation** in SystemVerilog

# J68 to Verilator

## Verilog top-level model

```
module j68_soc (
    input           rst_n,
    input           clk,

    input
uart_rxd,
    input
uart_cts_n,
    input
uart_dcd_n,
    output
uart_txd,
    output
uart_rts_n,
```

## Compiled to C++ by Verilator

```
verilator $TOP_FILE.v $COMPILE_OPT
$TRACE_OPT -top-module $TOP_FILE -exe
$CPP_FILES
cd ./obj_dir
make -j -f V$TOP_FILE.mk V$TOP_FILE
```

# Instantiation and simulation

## Instantiation

```
int main(int argc, char **argv, char **env) {
    Verilated::commandArgs(argc, argv);
    Vj68_soc* top = new Vj68_soc;
    while (tb_sstep < NUM_STEPS) {
        top->rst_n = (tb_sstep < (vluint64_t)24) ? 0 : 1;
        top->clk = top->clk ^ 1;
        top->eval(); if (Verilated::gotFinish()) break;
    }
    top->final();
}
```

## Add tracing

```
// in declarations
    // Init VCD trace dump
    Verilated::traceEverOn(true);
    VerilatedVcdC* tfp = new VerilatedVcdC;
    top->trace(tfp, 99);
    tfp->spTrace()->set_time_resolution("1 ps");
    tfp->open(file_name);

// in main loop
        top->eval(); tfp->dump(tb_time); if
(Verilated::gotFinish()) break; }
```

# Simulating in Verilog

## Feed 68k test program from simulated memory

```
reg  [15:0] r_mem_blk [0:(1 << ADDR_WIDTH) - 1];
reg  [15:0] r_q;

// 68k bus activty
  always@(posedge clock) begin
    r_q <= r_mem_blk[address][15:0];
    if (wren) begin
      if (byteena[0]) r_mem_blk[address][7:0]  <= data[7:0];
      if (byteena[1]) r_mem_blk[address][15:8] <= data[15:8];
    end
  end
  assign q = r_q;
```

# Simulating in Verilated C++

## Feed 68k test program from simulated memory

```
CART::CART(int w, bool debug, int size) …
    // Binary file loading
    void CART::load(const char *name, vluint32_t begin, vluint32_t end) …
    // Cycle evaluate
    void CART::eval(vluint64_t cycle, vluint8_t clk, // Cycle counter, clock
                    vluint8_t ce_n, vluint8_t oe_n, // Control signals
                    vluint32_t a, // Address
                    vluint32_t &q, vluint8_t &oe // Outputs: data, output enable
                   ) {
        if (!clk) return;
        a &= 0x7fffff;
        switch(width) {
          case 1:   // 16 bits
            a &= ~1; q = (mem_array[a] << 8) | mem_array[a + 1];
            break;
          case 2: // 32 bits
            a &= ~3; q = (mem_array[a] << 24) | (mem_array[a + 1] << 16)
                       | (mem_array[a + 2] << 8) | (mem_array[a + 3] << 0);
            break;
          default:
            q = mem_array[a];
        }
        oe = (!ce_n && !(oe_n & 1)) ? 1 : 0;
        oe |= (!ce_n && !(oe_n & 2)) ? 2 : 0;
    …
```

# SystemVerilog

## DPI (Direct Programming Interface)

```
import "DPI-C" function void dpi_trace_init();
import "DPI-C" function dpi_trace_fetch(
  input integer sr,
  input integer pc,
  input integer usp,
  input integer ssp,
  input integer lvl );
…
  always@(posedge clk)
  begin
    // Instruction fetch
    if (w_dbg_ifetch)
      dpi_trace_fetch({16'd0, w_dbg_sr_reg}, w_dbg_pc_reg,
w_dbg_usp_reg,
        w_dbg_ssp_reg, {29'd0, w_dbg_irq_lvl});
    …
  end
…
```

# 68k disassembler to DPI

- Connect the disassembler from Musashi to the DPI

```
#include "svdpi.h"
extern "C" {
#include « musashi/m68k.h"
}

void dpi_trace_fetch(int sr, int pc, int usp, int ssp, int lvl) {
    m68k_disassemble(dis_buff, prev_pc, M68K_CPU_TYPE_68010);
    fprintf(fh_j, "PC=%08lX  %s\n", prev_pc, dis_buff);
    fprintf(fh_j, "D0=%08X %08X %08X %08X %08X %08X %08X %08X \n",
                   regs[0], regs[1],  regs[2],  regs[3],  regs[4],
                   regs[5], regs[6],  regs[7]);
    …
```

# Musashi trace dump

## Musashi to same trace format

```
reg = m68k_get_reg((void *)NULL, M68K_REG_PC);
   m68k_disassemble(dis_buff, reg, M68K_CPU_TYPE_68010);
   fprintf(fh_m, "PC=%08lX  %s\n", reg, dis_buff);
   m68k_execute(1); // Musashi run 1 instruction
   fprintf(fh_m, "D0=");
   for (i = (int)M68K_REG_D0; i <= (int)M68K_REG_D7; i++) {
       reg = m68k_get_reg((void *)NULL, (m68k_register_t)i);
       fprintf(fh_m, "%08lX ", reg); }
```

## ...and diff traces

```
PC=000002F0  movea.l A0,A1
D0=11552299 33774401 00000000 00000000 00000000 00000000 00000000 00000000
A0=00008004 00008004 00000000 00000000 00000000 00000000 00000000 0000FFFC
USP=00000000 SSP=0000FFFC SR=2710 XNZVC=10000

PC=000002F2  move.l  #$09010101,(A1)+
D0=11552299 33774401 00000000 00000000 00000000 00000000 00000000 00000000
A0=00008004 00008008 00000000 00000000 00000000 00000000 00000000 0000FFFC
USP=00000000 SSP=0000FFFC SR=2710 XNZVC=10000
```

# to go a little further

## Links to open source projects

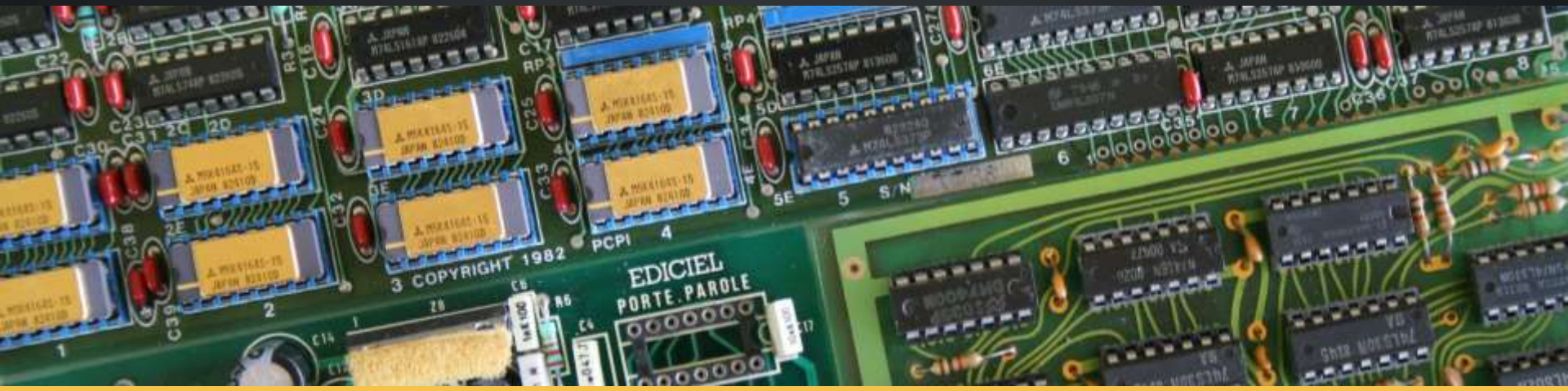
www.veripool.org/wiki/verilator
gtkwave.sourceforge.net
sio2sd.gucio.pl/wiki/English
www.codesrc.com/mediawiki/index.php/SCSI2SD
speedofmac.com/emulation/macifom.php
github.com/keirf/FlashFloppy/wiki
github.com/mist-devel/mist-board/wiki
github.com/MiSTer-devel/Main_MiSTer/wiki
github.com/Torlus/FPGAPCE
github.com/MiSTer-devel/TurboGrafx16_MiSTer
github.com/MiSTer-devel/Genesis_MiSTer

# download this presentation

## At silicium.org !

# * Contact us !

**And share your projects. Silicium action is based on popular support. As the technological heritage is getting bigger each day, Silicium's members are able to offer extensive knowledge in order to manage large-scale projects, up to the creation of a permanent museum.**




**Silicium.org**

+33(0)5.61.85.90.33

info@silicium.org

**You can also find us at:**

twitter.com/occitel

facebook.com/siliciumasso

Meet us Saturday At 11:00 !

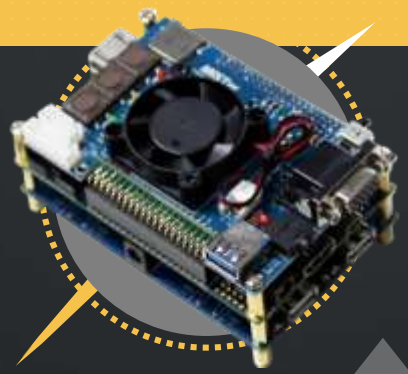
silicium

# Retrocomputer Preservation

**Preserving is not just about storing computers, disks and books in a hangar**

## FOSDEM Brussels, 2 & 3 February 2019


**Original machine ?**



**Hard Emulation ?**

**Soft Emulation ?**