

Virtual Circuit Bending

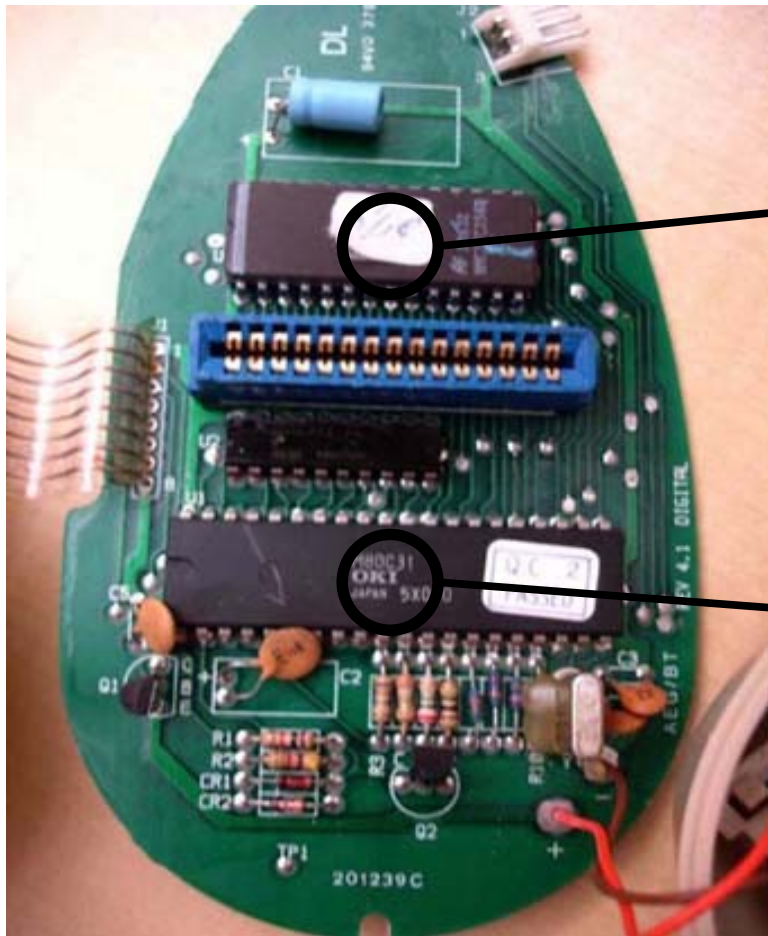
Taking circuit bending concepts
to places where wires can't go.

The idea



Virtual Circuit Bending @ Bent 2006

What about the ICs?



Static data (ROM)

CPU

How to gain access?

Can't fit wires, knobs, or dials in there!

Need a system that can mimic their behavior.

Software-based emulation is only reasonable option.

What is an emulator?

Systems have expected behaviors

- We would like to reproduce these behaviors in software.
 - Don't need the actual device

An attempt to reproduce the expected behavior of a system.

- e.g. game consoles running on desktop computers.

Two problems

1. Emulators are complicated.
2. Nearly all emulators are written for a desktop environment

What environment to use? (1)

- Requirements
 - Input flexibility
 - Keyboards, MIDI controllers, etc.
 - Audio framework
 - Graphics framework
- Flexibility is crucial

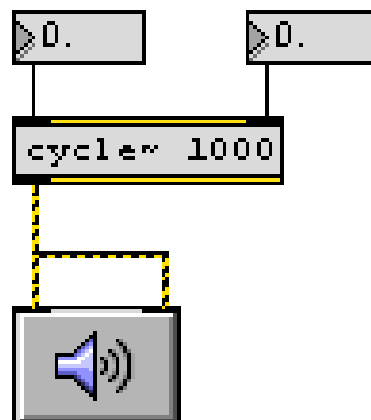
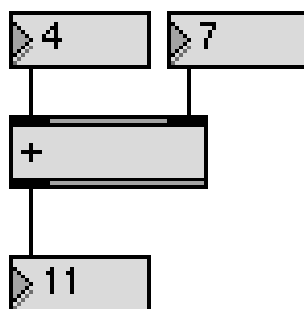
What environment to use? (2)

- Requirements
 - Input flexibility (Max)
 - Audio framework (MSP)
 - Graphics framework (Jitter)

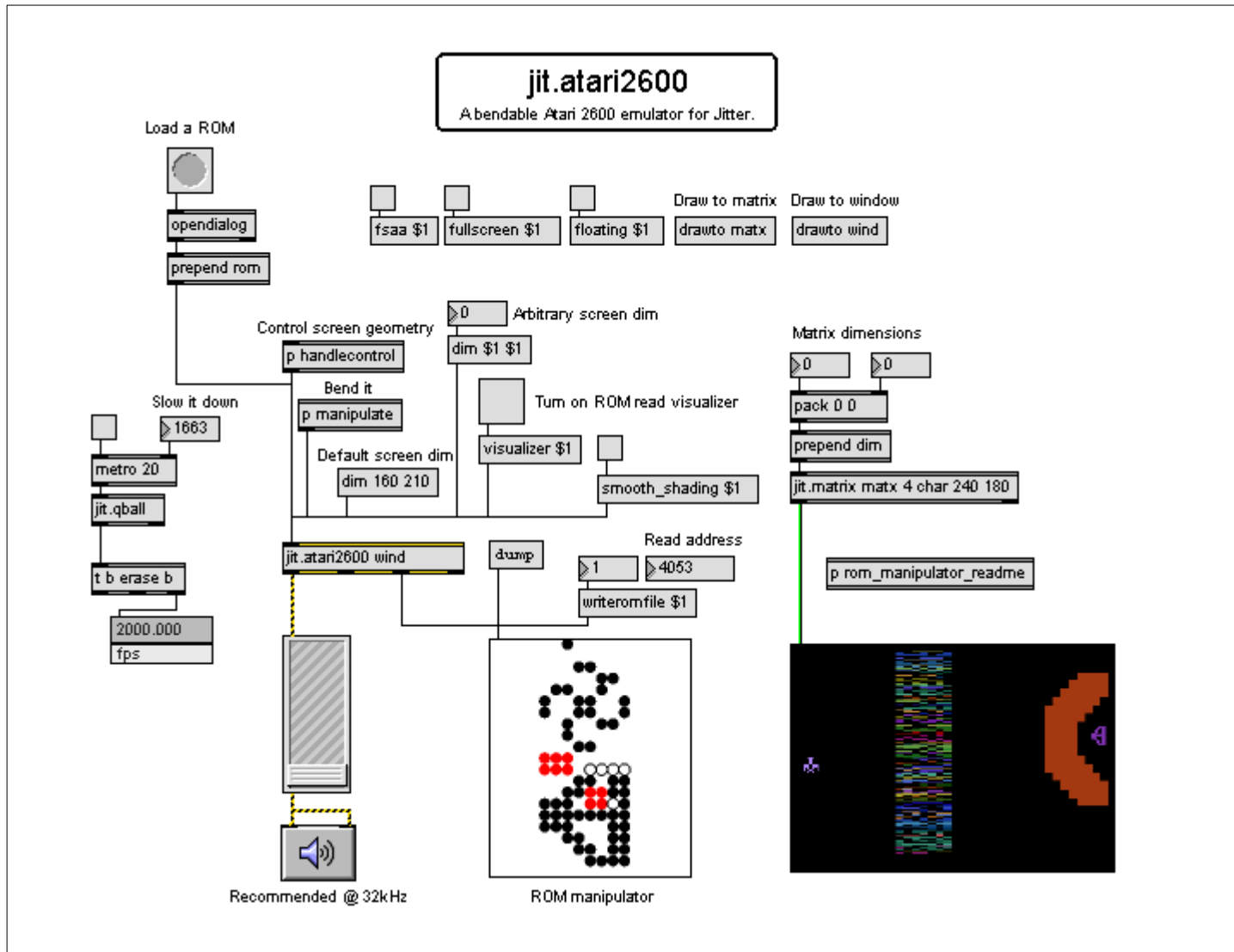
Max/MSP in a nutshell

Max is a graphical programming environment.

The *max object* is the central concept:



The Max patch



Externals

Custom Max objects can be written in C/C++.

- Objects of this type are called *externals*.

```
#include "ext.h"
#include "z_dsp.h"
#include "SP0256.h"

void *intellivoice_class;

void *intellivoice_new(void);
t_int *intellivoice_perform(t_int *w);
void intellivoice_dsp(t_intellivoice *x, t_signal **sp, short *count);
void intellivoice_in1(t_intellivoice *x, long value);

void main(void) {
    setup((t_messlist **) &intellivoice_class, (method) intellivoice_new, (method) dsp_free, (short) sizeof(t_intellivoice), 0L, 0);
    addinx((method) intellivoice_in1, 1);
    addresss((method) intellivoice_dsp, "dsp", A_CANT, 0);
    dsp_initclass();
}

void *intellivoice_new(void) {
    t_intellivoice *x = (t_intellivoice *) newobject(intellivoice_class);
    dsp_setup((t_pxobject *) x, 0); // no inlet
    outlet_new((t_pxobject *) x, "signal"); // outlet
    intin(x, 1);
    return (x);
}

void intellivoice_dsp(t_intellivoice *x, t_signal **sp, short *count) {
    dsp_add(intellivoice_perform, 3, sp[0]->s_vec, sp[0]->s_n x);
}

void intellivoice_in1(t_intellivoice *x, long value) {
    resetProcessor(x);
}

t_int *intellivoice_perform(t_int *w) {
    t_float *outL = (t_float *) (w[1]);
    t_intellivoice *x = (t_intellivoice *) (w[3]);
    tick(x, outL, (int) (w[2]));
    return (w + 4);
}
```



intellivoice~

Three types of bending

Abstraction modification:

The expected behavior of the system is abstracted and often clearly represented in the source code.

Processor modification:

Direct modification of the virtual processor.

Instruction modification:

Can produce desirable results (for simple processors.)

Abstraction – an example

Intellivoice emulator

```
typedef struct _intellivoice
{
    char filterParamA;
    char filterParamB;
    ...
} intellivoice;

void decode(intellivoice* x)
{
    char instruction = readByte();

    switch (instruction) {
        case 0x8:
            START();
            break;
        case 0x4:
            //Load filter parameters

            x->filterParamA = readByte();

            x->filterParamB = readByte();
            break;
        ...
    }
}
```

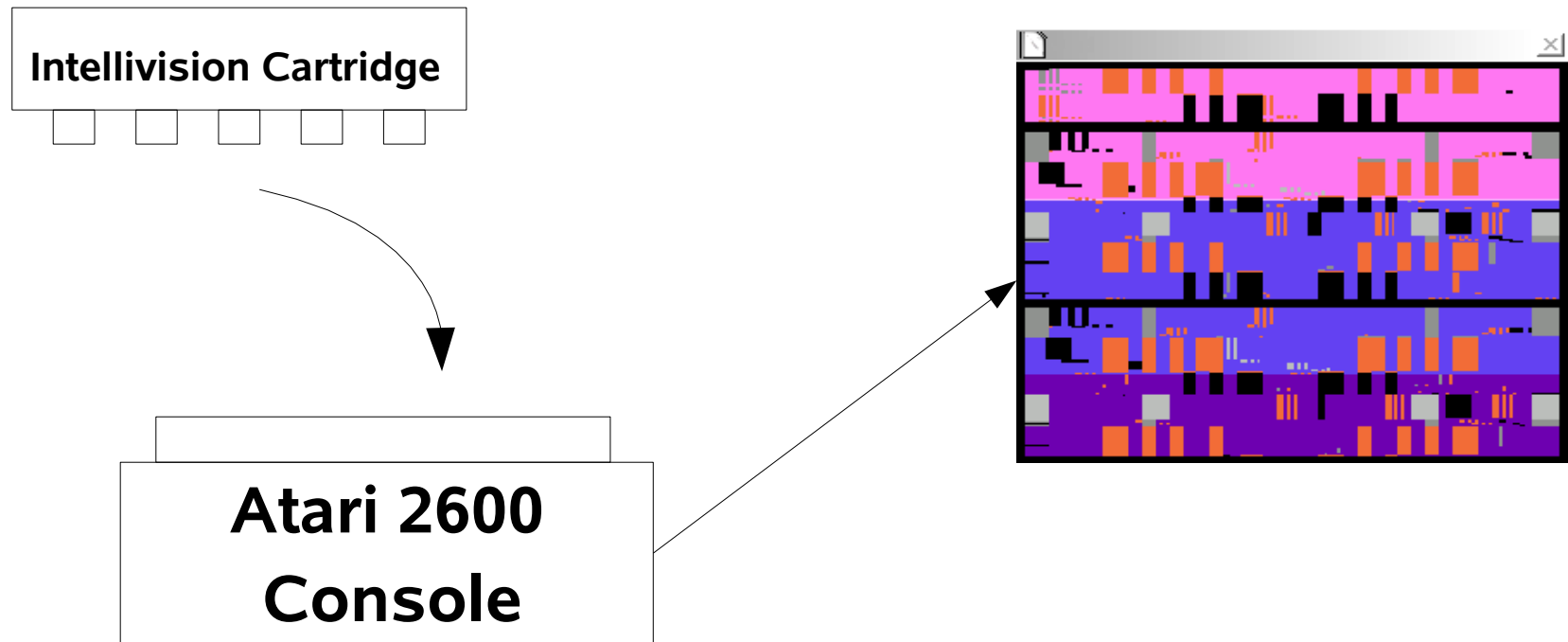
Another abstraction example

6502 Emulator

(changed via Max)

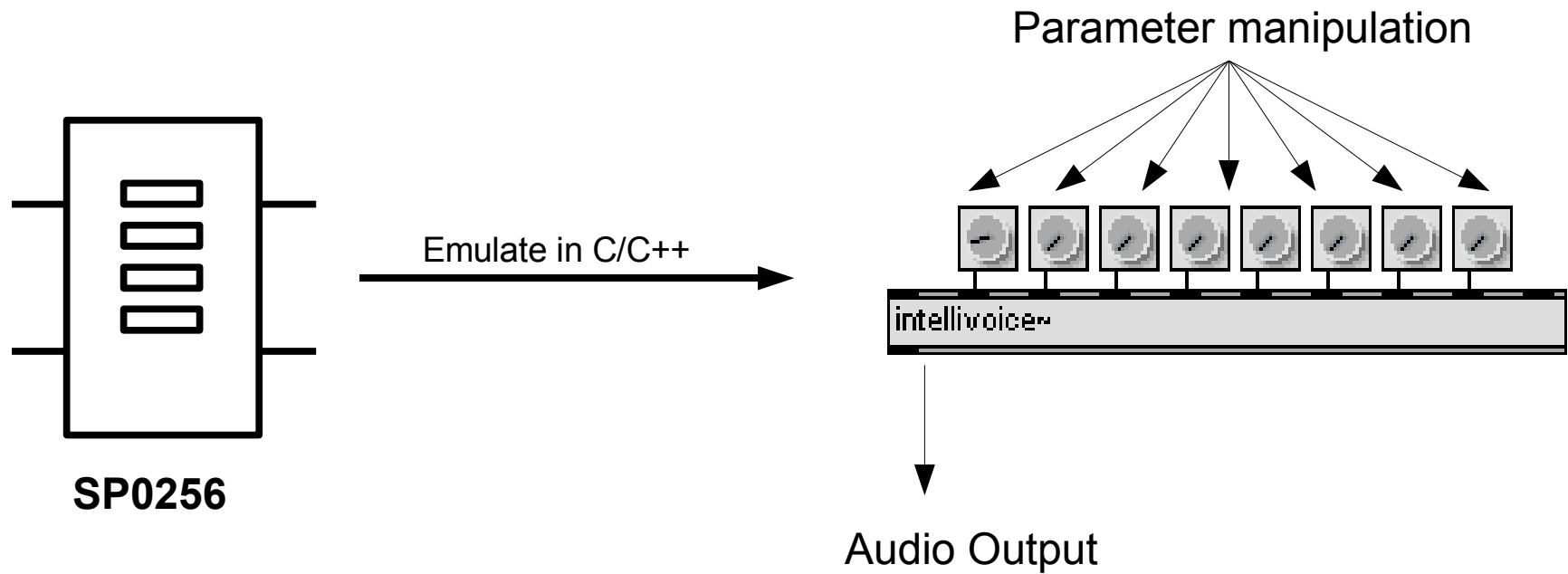
```
switch(addr)
{
  ...
  case 0x1B: // Graphics Player 0
  {
    // Set player 0 graphics register
    myGRP0 = value;
    if(modifierValue != 0)
    {
      myGRP0 = modifierValue;
    }
    ...
  }
}
```

Instruction modification



Proof of concept

Intellivoice voice synthesis peripheral:



A peek inside

Max input handling

```
void intellivoice_in1(t_intellivoice *x, long value)
{
    x->filterParamA = value;
}

void intellivoice_in2(t_intellivoice *x, long value)
{
    x->filterParamB = value;
}
```

Intellivoice emulator

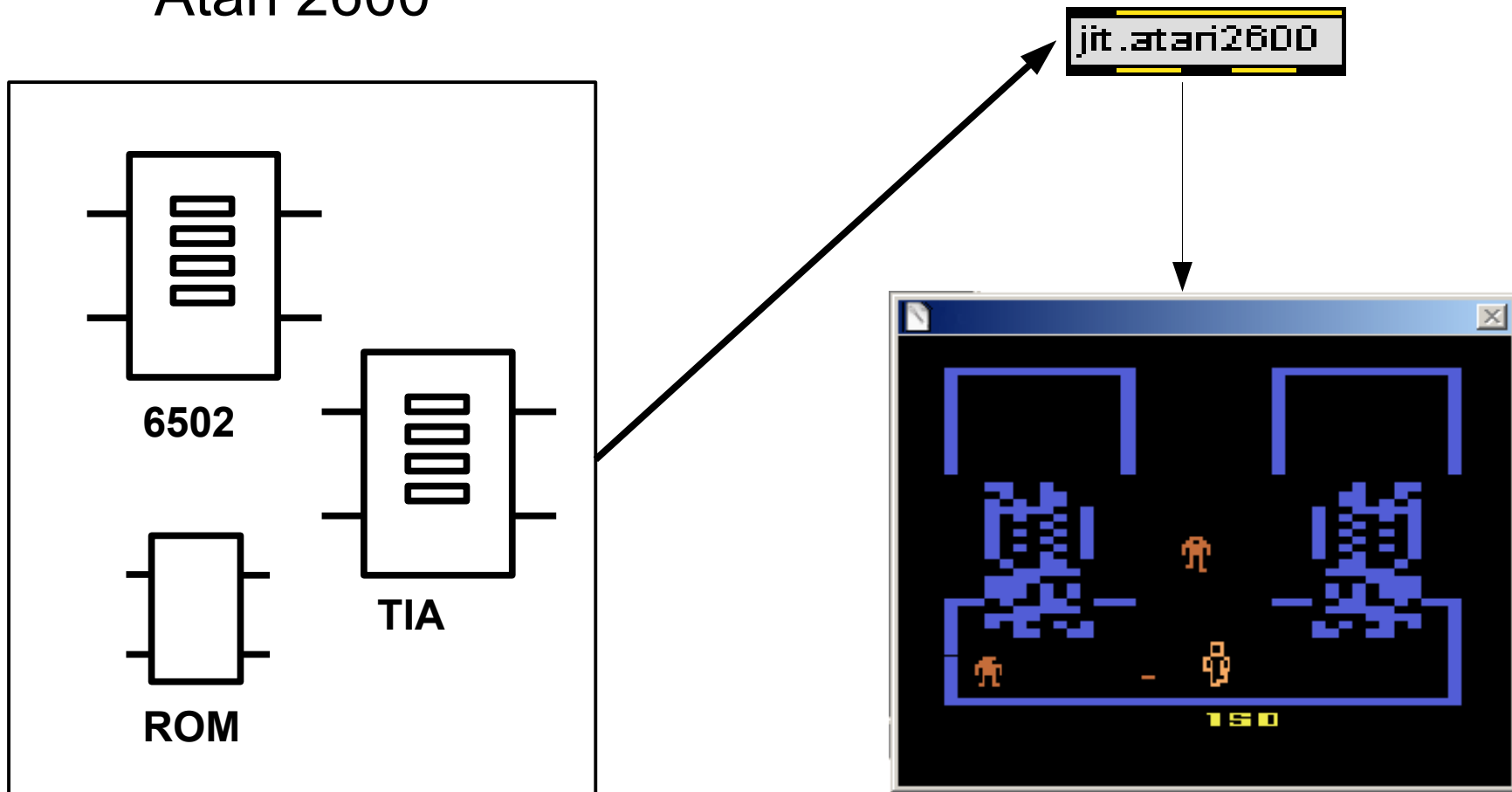
```
typedef struct _intellivoice
{
    char filterParamA;
    char filterParamB;
    ...
} intellivoice;

void decode(intellivoice* x)
{
    char instruction = readByte();

    switch (instruction) {
        case 0x8:
            START();
            break;
        case 0x4:
            //Load filter parameters
            x->filterParamA = readByte();
            x->filterParamB = readByte();
            break;
        ...
    }
}
```

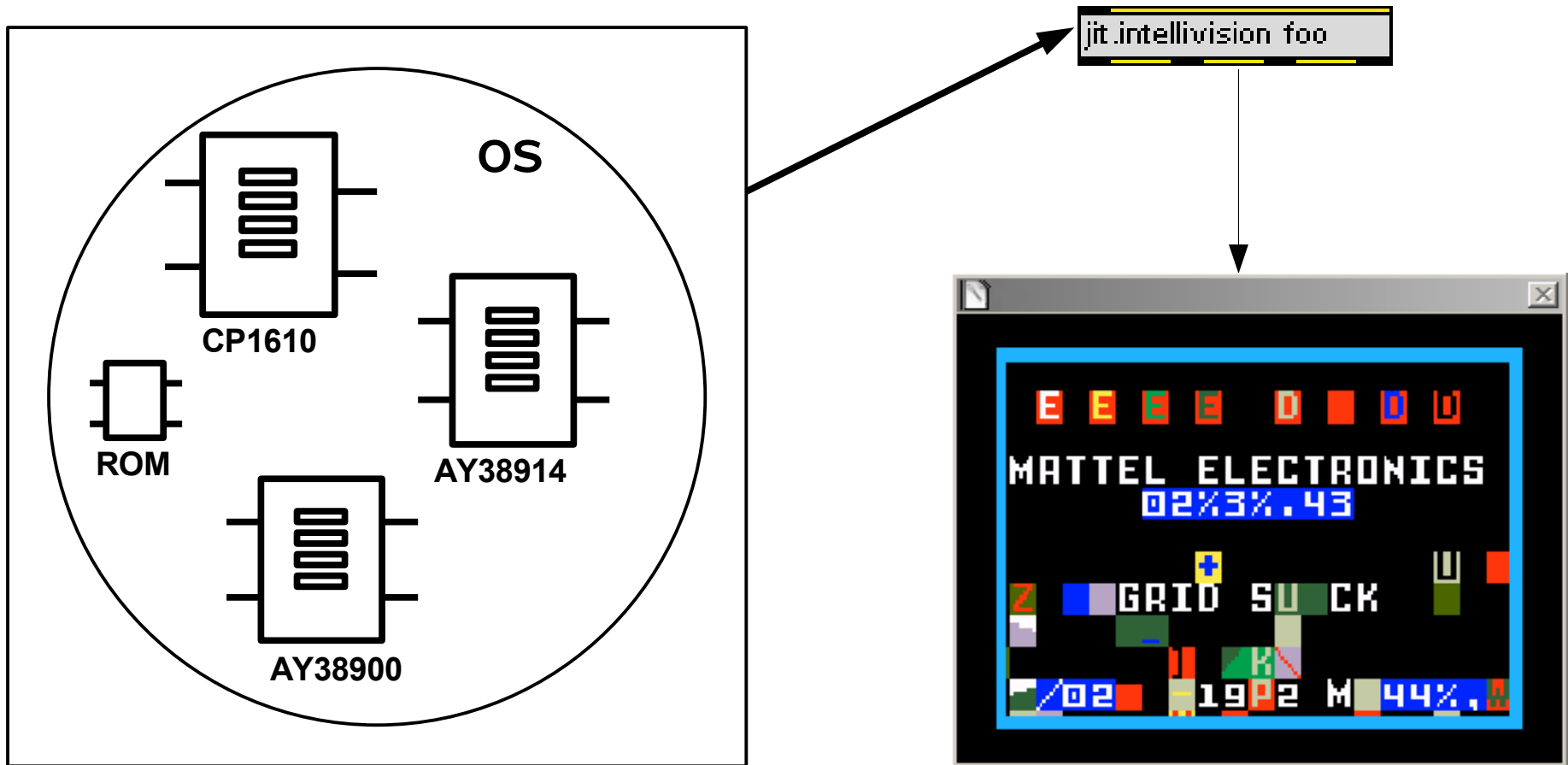
Extending the idea

Atari 2600



Hitting some barriers

Intellivision



Rolling your own

Choose a system, find the source

- Start simple, data/audio only

Turn it into an external

- Modify existing SDK examples

Bend

Some suggestions

The search space is huge. Can feel daunting.

Experimentation **always** pays off.

Contact

www.mmonoplayer.com

info@mmonoplayer.com