Preserving History with KiCad
Portable Apple II

Use the minimum number of chips necessary for a portable Apple II to run games.
(Okay, Oregon Trail.)

Original Apple II used 88 chips
The Trinity

The most popular computers for almost a decade.

**TRS-80**
Radio Shack / Tandy

**Apple II**
Apple Computer

**Commodore 64**
CBM

Credit: Wikimedia Commons, Various
Apple IIGS

From Wikipedia, the free encyclopedia

The Apple IIGS (styled as IIGS), the fifth and most powerful of the Apple II family, is a 16-bit personal computer produced by Apple Computer, Inc. While featuring the Macintosh look and feel, and resolution and color similar to the Commodore Amiga and Atari ST, it remains compatible with earlier Apple II models. The "GS" in the name stands for "Graphics and Sound," referring to its enhanced multimedia hardware, especially its state of the art audio.¹

The microcomputer is a radical departure from any previous Apple II, with its 65C816 16-bit microprocessor, direct access to megabytes of RAM, and mouse. It was the first computer produced by Apple to use a color graphical user interface (color was introduced on the Macintosh six months later) and Apple Desktop Bus interface for keyboards, mice, and other input devices. It is the first personal computer to have a wavetable synthesis chip, utilizing technology from Ensoniq.

The IIGS set forth a promising future and evolutionary advancement of the Apple II line, but Apple increasingly focused on the Macintosh platform. The IIGS clock speed was intentionally limited below the maximum for the 65C816 so the system would not outperform the Macintosh.² The IIGS outsold all other Apple products, including the Macintosh, during its first year in production.³ Citation needed Apple ceased IIGS production in December 1992.

1 Hardware features
   1.1 Graphics modes
   1.2 Audio features
   1.3 Expansion capabilities
   1.4 Development and codenames
2 Release
   2.1 Limited Edition ("Woz"-signed case)
   2.2 Influence on later computers
   2.3 Upgrading an Apple IIe
3 Software features
   3.1 8-bit Apple II compatibility
   3.2 System Software
      3.2.1 Graphical user interface
      3.2.2 Extensibility

Manufacturer Apple Computer, Inc.
Release date September 15, 1986; 32 years ago
Introductory price US$999 (equivalent to $2,283 in 2019), excluding monitor
Discontinued December 1992
Operating system Apple ProDOS
CPU 2.8 MHz 65C816
Memory 256 kB or 1 MB (expandable up to 8 MB)
Graphics VGA, 128 colors, 320 × 200
Apple II

CPU

RAM

ROM

CPU
Apple IIc Logic Board
Apple IIgs

16-Bit / 8 MB RAM / 4096 Colors

Apple II Backward Compatibility
Why Bother?

Aren’t PDFs Good Enough?
Platform Requirements

- Platform
- Open File Format
- Long Term Support
- Accessible
- Scripting
- Autorouter
Cross Platform
Open file format

KiCad

EAGLE

Altium
Long Term Support

- Introduction
- Growth
- Maturity
- Decline
Accessibility

Are others open and willing to collaborate on the platform

Cost

Usability

Capability
Scripting

```python
from boms_away import sch, datastore
from boms_away import kicad_helpers as kch
from boms_away import exportpluging as export_plugins
from boms_away import plugin_loader

class DBPartSelectorDialog(wx.Dialog):
    def __init__(self, parent, id, title):
        wx.Dialog.__init__(self, parent, id, title)
        self.selection_idx = None
        self.selection_text = None

        vbox = wx.BoxSizer(wx.VERTICAL)
        stline = wx.StaticText(self, -1, 'Please select from the following components')
        vbox.Add(stline, 0, wx.ALIGN_CENTER|wx.TOP)
        self.comp_list = wx.ListBox(self, -1, style=wx.LB_SINGLE)
        vbox.Add(self.comp_list, 1, wx.ALIGN_CENTER_HORIZONTAL|wx.EXPAND)
        self.SetSizer(vbox)
        self.comp_list.Bind(wx.EVT_LISTBOX, self.on_selection, id=wx.ID_ANY)

    def on_selection(self, self, event):
        self.selection_text = self.comp_list.GetStringSelection()
        self.selection_idx = self.comp_list.GetSelection()
        self.Close()

    def attach_data(self, self, data):
        list(map(self.comp_list.Append, data))
```

```python
output("/tmp/ulp_output.txt","at") {
    if (argc != 5 & argc != 3) {
        digMessageBox("Incorrect number of arguments","Ok");
        exit(1);
    }
}

real width = strtol(argv[1]);
real height = strtol(argv[2]);
real x = 0;
real y = 0;
if (argc >= 4) {
    x = strtol(argv[3]);
    y = strtol(argv[4]);
}

string s = "";
sprintf(s,"WIRE (%f %f) (%f %f) (%f %f) (%f %f) (%f %f);",
    x+width/2.0,y+height/2.0,
    x+width/2.0,y-height/2.0,
    x-width/2.0,y+height/2.0,
    x-width/2.0,y-height/2.0
);
exit(s);
}
```
Autorouter
KiCad is Best Option
Not all Pin 35s are created equal
**Long Wire**

Useful for PDF or static schematics since there is limited search capability. However, it means a lot of manual tracing.

**Pin Labels**

Cleaner schematic and easier to do automated search. Potential for typos.
How to automate creating decoupling capacitor arrays?

(Reverse) SKIDL?
Project for capturing vintage, classic, aka old computer schematics in KiCad.

- **1 commit**
- **1 branch**
- **0 releases**
- **1 contributor**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Status</th>
<th>Commit</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amiga</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
<tr>
<td>Amstrad</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
<tr>
<td>Apple</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
<tr>
<td>Atari</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
<tr>
<td>Commodore</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
<tr>
<td>Sinclair</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
<tr>
<td>Tandy-RS</td>
<td>Initial</td>
<td></td>
<td>28 minutes ago</td>
</tr>
</tbody>
</table>

**readme.md**

**Bit Preserve**

Recreating classic computer schematics. Let's convert all those random scanned PDFs into a modern, editable, and re-usable format.
Visit GitHub Repository #3

- CVA-Mem-1.sch
- Capacitors-6.sch
- U1gS Schematic.kicad_pcb
- U1gS Schematic.pro
- U1gS Schematic.sch
- IWM-4.sch
- MEGA-II-2.sch
- RAM-9.sch
- Slots-7.sch
- Sound-8.sch
- UG12-5.sch
- Video-3.sch

Latest commit 1649ac 36 minutes ago
Collaboration

1. Pick a system
2. Initial Capture
3. Submit Rev
4. Validate Capture
5. Verify with Hardware
6. Publish Final

github.com/baldengineer/bit-preserve
Contact Me

james@baldengineer.com
e-mail | www | twitter

github.com/baldengineer/bit-preserve