CPLD Modules

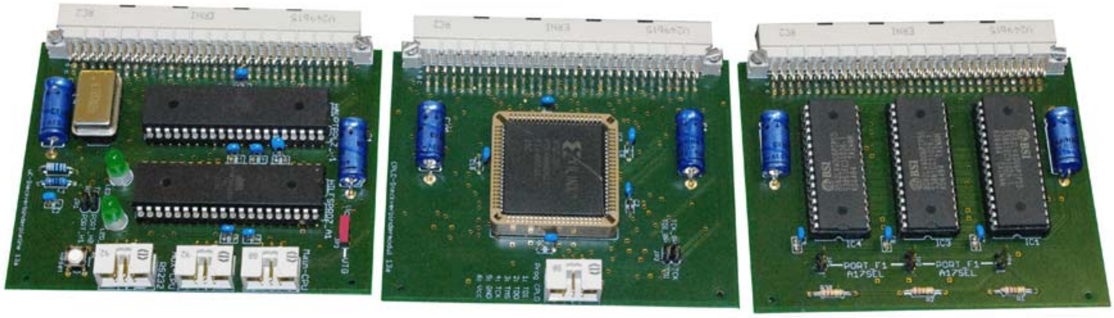
March 2014

Complex FPGAs are terrible to design with. It is way too much for newcomers. Basic research is hampered as well. To investigate architectural principles, one needs a suitable environment to experiment. Computer simulation is too slow, FPGA implementations pose their own intricate problems like routing, clock delays and so on – especially when uncommon circuitry is to be implemented (the manufacturer's coding recommendations provide well for NIOS processors, ARM cores, FIFO buffers, digital filtering and the like, but obviously not for ReAl machines...).

Hence we revive a venerable principle of hardware design. Instead of using a large FPGA, we implement the logic in comparatively small CPLDs (and some auxiliary modules) which are inserted in a backplane and connected by wire (wire-wrap technology).

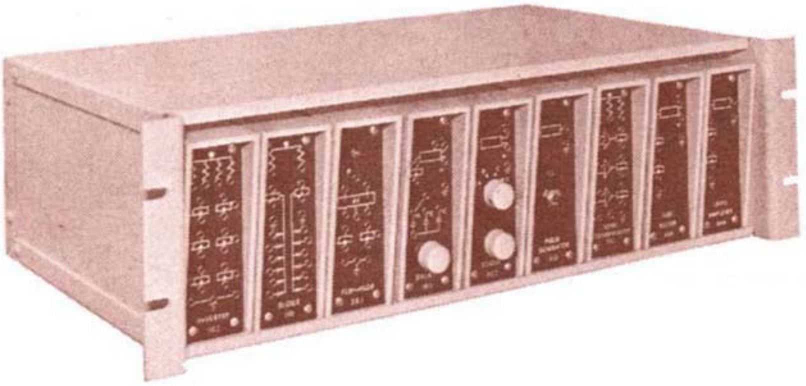
It is basically a revival of the old modular plug-in principle. However, what was a card with many digital circuits in bygone times, is now a CPLD.

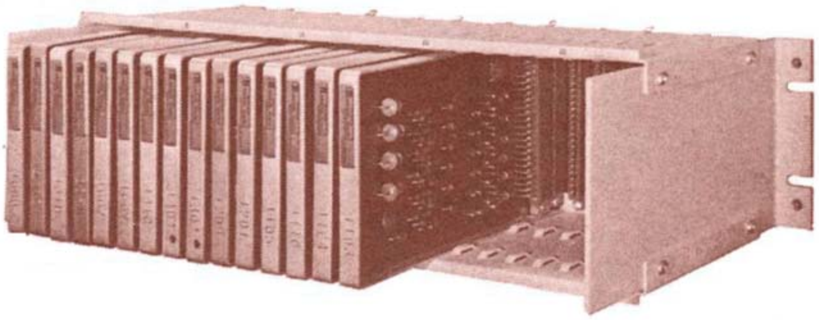
Thus, processors, programmable logic controllers and the like can be designed the traditional way. This also applies to bring-up and troubleshooting. It can be done with real scopes and logic analyzers instead of complex simulation software... Design bugs are corrected by reprogramming CPLDs and changing wire-wrapped connections.



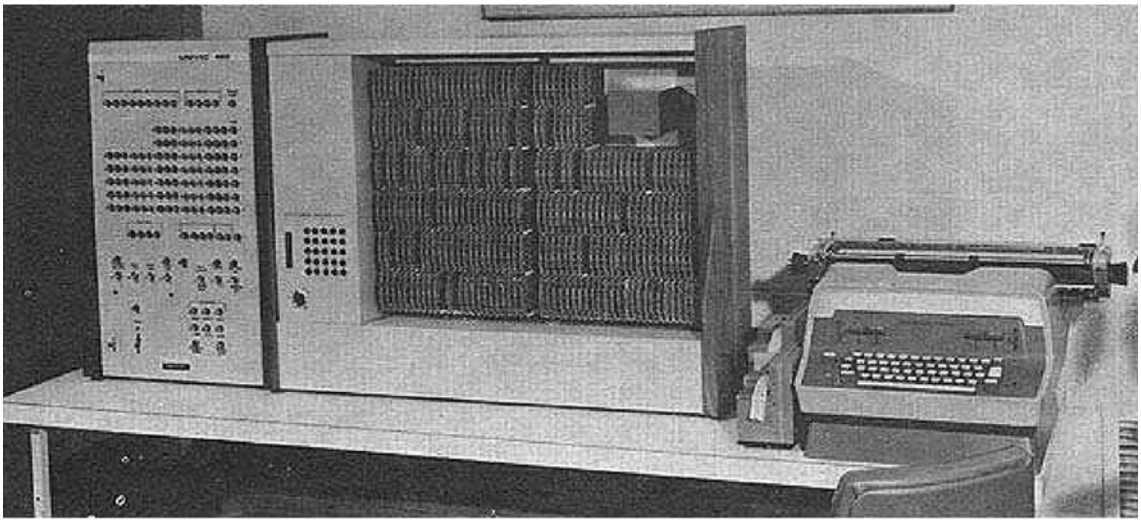
From left to right: service processor (SVP) module (containing two Atmel Atmega microcontrollers), CPLD module (Xilinx 95108), SRAM module (128k • 24 or 512k • 24).

Legacy modules (DEC):



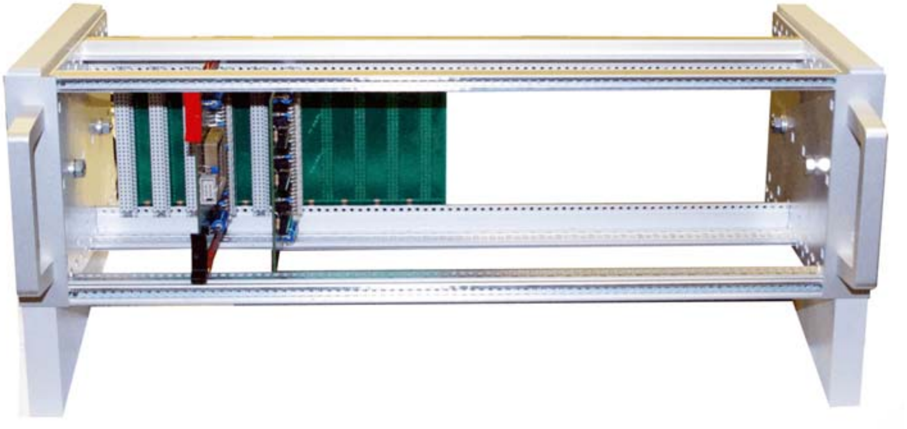


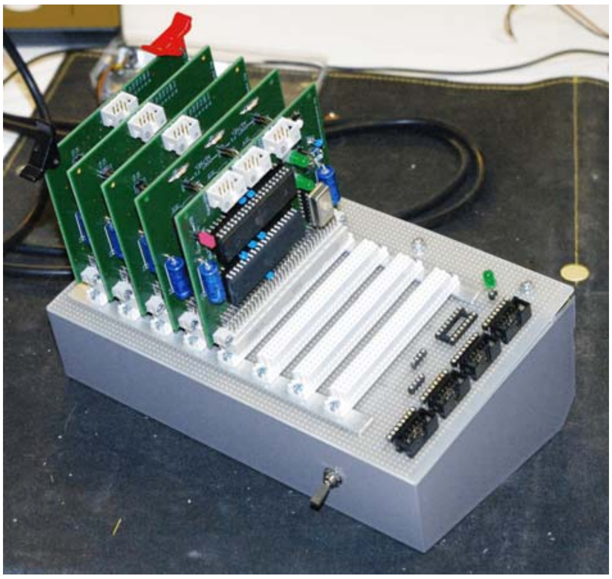
The Univac 422 Training Computer (UNIVAC Divison of Sperry Rand Corp.):

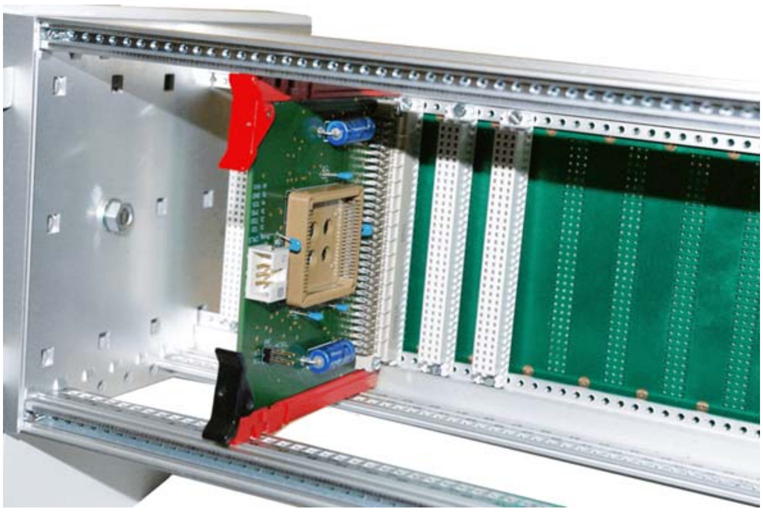


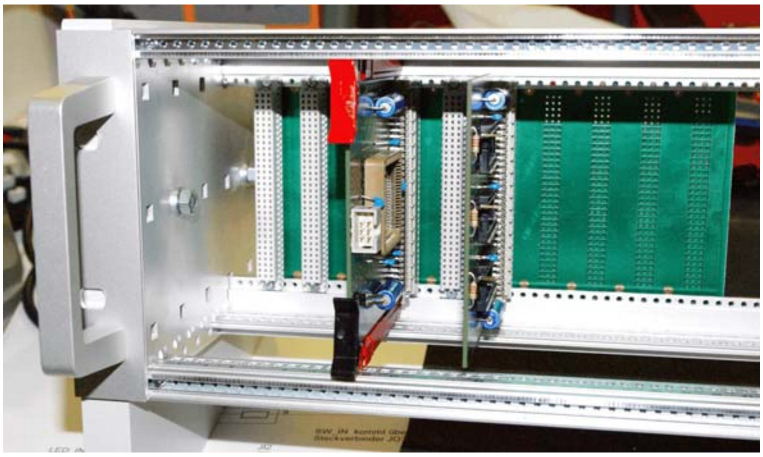
It consists of 1176 transistors and 4639 diodes. The machine word is 15 bits long. The memory capacity is 512 words. In a rough estimate, the logic circuitry corresponds to somewhat 3000 NAND gates or 800 7400 TTL circuits. Five Xilinx 95108 CPLDs should be sufficient...

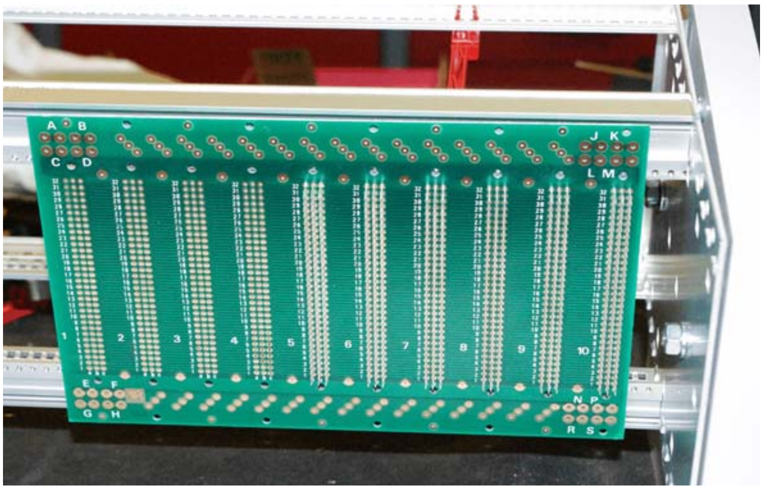
The form factor of our modules is 19", 3 U, half length of an eurocard. Thus we can rely on readily available 19" components.










A
B
C
D
J
K
L
M
1
2
3
4
5
6
7
8
9
10
E
F
G
H
N
P
R
S