

# SoftWire

Aaron Wohl - N3LIW

*SoftWire is a system for designed schematics using digital logic. The schematics are translated into assembly code for a target microprocessor.*

A friend in the hospital had been a radio man in WW II. I built him a get well card with a single chip cpu that played a get well message. Later, I changed the program to become a useful tool to myself. The program I inserted was the Morse code alphabet which changed my get well message into a single chip morse code tutor. It became an easy learning tool for me as well as a conversation piece.

Several months later I came across an article in a ham magazine for a much simpler circuit that needed four TTL chips. Why did they use four chips -- when one would do?

I asked local hams how they designed and built logic/controller circuits. All but one used TTL chips and their general census was any kind of assembly programming was beyond them. However, the individual ham operator attempting to program a microcontroller was genuinely interested in/with the idea but unable to get anywhere. He felt his lack of progress was due to the fact that ham radio exams and study manuals often speak of logic gates but do not mention microcontroller instructions.

**SoftWire** is a digital logic schematic drawing program. It translates the schematic into assembly instructions for a target microprocessor or PAL assembler. **SoftWire** is currently under development and is being written in **Smalltalk/V** for Microsoft Windows. The initial target architecture is the Microchip **PIC16Cxx** series.

**SoftWire** provides icons for:

- Traditional logic blocks, AND, OR, flip-flops (D, RS, JK)
- Finite state machines
- Time meters to measure delays through parts of the schematic.

**SoftWire** schematics can be annotated with a 'time meter'. **SoftWire** updates this display to indicate the delay a signal experiences from one part of the circuit to another. For example:

- An output pin must be updated  $20\mu\text{s}$  after changes in any of the input pins. Then a meter icon is connected to the input pins and output pins. It is set to read in micro seconds where the meter then is updated after code generation.

- This is analogous to the way word processors update page break indicators after printing or repaginating.
- If the delay shown on a meter is larger than real time constraints allow, a constraint may be set on the meter to ask for a delay less than or equal to a given time. The **SoftWire** compiler will try harder to speed up that path, perhaps at the cost of increased code size.

Constraints may also be set for minimum and maximum delay so as to produce a constant response time for a path.

- The meter metaphor provides a powerful tool to deal with real time constraints. Traditional high level compilers have optimization settings on a per file basis. However, they cannot express specific constraints for certain paths to be fast enough, and cannot express the concept of making some execution paths arbitrarily slow.

The **SoftWire** schematic as a visual program metaphor is attractive for logic replacement for end users and is more hardware oriented. Some problems however, still remain. **SoftWire** currently thinks of each 'wire' in the schematic as carrying a bit from a single output to some number of inputs.

- A tri-state IO pin icon currently has an input connector dot, an output connector dot, and a direction connector dot. This does not look very natural in reading a schematic.
- The Microchip **PIC16C71** has an **onboard** analog to digital converter. **It** would be more natural to deal with the output of the converter as an eight bit integer flowing through a pipe rather than as eight wires.

The **SoftWire** program is available from the **CompuServ** ham radio sig, software library. **It** is also available on the internet from host **akutaktak.andrew.cmu.edu** [128.2.35.1] in the **/aw0g** directory. The author can be contacted via internet mail as **n3liw+@cmu.edu**.