Easier Asynchronous Chip Design

April 2025

Ron Lavallee
You Know Solutions LLC
Belgrade, ME USA
ronl@youknowsolutions.com

Key Terms – FloPro: Microsoft DOS based PC industrial control software
Flowpro Software: Windows based PC hobbyist control software
Flowpro Machine: Patented asynchronous parallel hardware implementation of Flowpro Software

Before considering a new asynchronous design flow one might ask "why hasn't asynchronous Chip design become mainstream?" Microsoft Copilot and OpenAI ChatGPT [1] responses are basically the same: Complexity, Tools and Support, Standardization, Awareness & Performance, Cost & Efficiency Trade-Offs and Market Inertia. It's a very reasonable assessment by AI that's easy to agree with. Any fresh asynchronous design flow that replaces the existing design flow would need to address all of these assessments.

I propose that research communities look into combining parallel systems design and asynchronous circuits as a research area. Task parallel systems is another way to reduce power in chips because tasks only run when they are needed. When implementing massively parallel systems, asynchrony has advantages over synchrony.

Parallel asynchronous systems have been slow to arrive probably because they are difficult to program and hard to build. Implementing a one million task parallel architecture chip using a synchronous design will be daunting. Clock routing and gating in particular along with functional decomposition will be very hard. On the other hand you might say that asynchronous implies parallel as well as clock-less. Asynchronous design is a natural fit for parallel systems chip design. There is a big advantage to not using a clock with massively parallel systems. Power savings would be significant but alone it won't be enough. A software and hardware system is needed to take the advantages of a parallel architecture and asynchronous circuit design to easily build lower power systems. That system is called a Flowpro Machine, a parallel asynchronous propagation machine.

The AI assessment asserts what many believe, asynchronous Chip design is just harder to do and managing all of the handshaking is one of the reasons. A Flowpro Machine is an event model that is built with parallel decision flowcharts using asynchronous block circuits. With the combination asynchronous circuits and a Flowpro Machine the 'value proposition' now becomes; lower power, easier design flow and true parallel execution.

A Flowpro Machine is a propagation flow based computational machine and a common software and hardware approach to parallel asynchronous chip design and implementation. Flowpro machines support both state-less and state-full design models. Design entry is via parallel, hierarchical decision flowcharts (Fig. 1) that are synthesized to parallel, asynchronous hierarchical flowchart circuit constructs. Each flowchart circuit construct is traceable to a design entry flowchart block. Flowpro Software is used for design entry of parallel flowcharts. The flowcharts are compiled to an object file which can be downloaded to a Turing machine for multitasking (concurrent) execution or compiled and synthesized for an FPGA (Field Programmable Gate Array) or in the future an FPFA (Field Programmable Flowpro Array).

A Flowpro Machine works by propagating multiple decision flowcharts simultaneously and starting functions as the parallel propagations proceed. The graphic decision flowcharts (Fig. 1) consists of 2D Enable, Control and Decision blocks which are low-level symbols that can be combined and encapsulated into higher-level 3-D symbols called Action, Test and Task objects. A 3-D flowchart object (Fig. 1) indicates that there are additional flowchart structures behind the 3-D object. A Flowpro Machine project is an assembly of parallel flowcharts and Task objects constructed in a hierarchical fashion. After constructing the flowcharts the blocks and objects are ordered (numbered) according to the flow lines with an algorithm that attempts to order all blocks higher than any block leading to it. Ordering defines all of the Atomic Paths on all flowcharts. An Atomic Path is a sequence of only ascending numbered blocks. After the system throughput time has been determined an Atomic Time parameter can be applied to guarantee throughput requirements for a Flowpro Machine system. Atomic Time is defined as the maximum time allowed to propagate an Atomic Path. When a flowchart is enabled, i.e. started, a propagation signal flows out of the Enable Block and follows the flowline. The leading edge of this signal triggers flowchart block functions without stopping as they are encountered. Once a flowchart begins propagating it is always propagating somewhere until the flowchart is disabled. Spiking signals (Fig. 3) are used to initiate and terminate flowchart functions (Start, Abort, Outputs, etc.).

In a Flowpro Machine, Action Objects and Test Objects are referred to as Recall Objects and each represents an encapsulation of a portion of a flowchart (Fig.1). Recall

objects do not contain any flowline loopbacks within the object encapsulation. Placing a Recall object on a flowchart is equivalent to expanding the contents of the Recall object in the flowchart between the object entry and exit points. A Task Object, also referred to as a Reason Object, can have flowline loopbacks and is equivalent to a flowchart in construction. A Task Object cannot be the top level of a Flowpro Machine project hierarchy. A flowchart is always the project top level of the project hierarchy. Task objects can be controlled and monitored from flowcharts and objects. The Task Object controls (Fig.3) are: 'Start' (begin, resume operation after a stop), 'Restart' (begin, resume operation from the beginning of the Task Object), 'Stop' (pause operation), 'Abort' (disable task operation), and monitoring is: 'Done' (task complete) and 'Running' (task operation enabled). When a task object is 'Done', the calling flowchart or Task Object has the option to abort or not to abort the called Task Object. When a Task Object is 'Done', all task objects called by this Task Object will continue to run. When a Task Object is aborted, all task objects called by this task object will be aborted. This is an automatic function that is handled by hardware and a signal called Hierarchy Abort. A flowchart that becomes disabled will abort all of the Task Objects in that flowchart's project hierarchy. This is a powerful and simple means to control when various portions of circuitry are idle or running, thereby saving power.

A demonstration software utility for converting Flowpro Machine flowcharts to Verilog code for FPGA execution has been written. The utility (FM2VU) provides a basic Verilog design of flowchart blocks and objects and their hierarchy of execution. Flowcharts that comprise a Flowpro Software project are compiled and downloaded to disk. Compiling the flowcharts ensures that the project is error-free and ready to run. The utility then reads this file and translates it to Verilog code. The FPGA manufacturer's development system is used to synthesize (Fig. 2) and download to the FPGA chip. Flowpro Software and Flowpro compatible MCU and I/O can be used for application behavioral debug.

A good example of a Flowpro Machine that shows hierarchy and power saving is a parallel 8-bit adder (Figs 1 - 3). The 8 bit adder flowchart circuitry is defined by a Task Object called "FM – 8BA 8-Bit Adder". A Flowpro Machine project was created using Flowpro Software and saved to disk. This disc file was then used by the FM2VU to produce Verilog code that was manually loaded into an Intel Quartus development system and downloaded to an Altera Cyclone V FPGA for demonstration. The adder consists of 19 parallel propagating tasks, one flowchart and 18 Task Objects. Adding more bits to the addition width would only require adding more parallel flowcharts and have very a little effect on throughput. All 4 bit partial-sums (carry in = 0 and = 1) are calculated in parallel. Each partial-sum Task Object flowchart 'Start' an individual Task Object flowchart indicating ON status for each bit of the four per partial-sum. When the partial-sum tasks are 'Done', the status of the carry outs for bits 0-3 is known and the partial-sums that are not valid for the current addition are aborted. A read command is generated and processed only by the ON Task Object flowcharts that are running.

The fact that the Flowpro Machine discussed above is patented should not hold back Flowpro propagation technology. A Flowpro Machine falls under the US Patent 10,181,003, "Processing Circuits for Asynchronous Modeling and Execution". The patent defines an asynchronous propagation computational machine that is decision graph based with common software and hardware function processing. So how can academia do research into a Flowpro Machine? US Patent 10,181,003 is a US only patent meaning there can't be claims against those developing, researching or using the patented technology outside of the United States. For those within the United States and wish to do research into the patented technology we can provide a document that releases any intellectual property claim against a researcher or Institute, as long as there isn't any commercialization of devices within the United States that incorporate the patented technology. The document further states that a researcher or Institute will seek a license before commercializing Flowpro Machine devices. Our intent is to foster the infrastructure that is needed to advance propagation flow technology.

Is a Flowpro Machine (FM) its' own computational machine? I believe it is, but a mathematical proof does not exist, yet. A Flowpro Machine can implement a Turing Machine but a Turing Machine can only simulate a Flowpro Machine. If you are a researcher or someone interested in doing a 'deep dive' and a Flowpro Machine paper, we will support you. We will provide the FP2VU, Flowpro Software and our expertise to help you with your evaluation. I am hoping that this paper will enlighten and foster research into asynchronous propagation machines and specifically a Flowpro Machine.

If you always do what you've always done, you'll always get what you always got!

C5 Corvette development team

[1] Microsoft Copilot & OpenAI ChatGPT, "Why hasn't asynchronous chip design become Mainstream?", January 22, 2025

[2] US Patent 10,181,003, "Processing Circuits for Parallel Asynchronous Modeling and Execution", January 15, 2019, Ronald J Lavallee, Thomas C Peacock

[3] You Know Solutions LLC, https://youknowsolutions.com, documentation

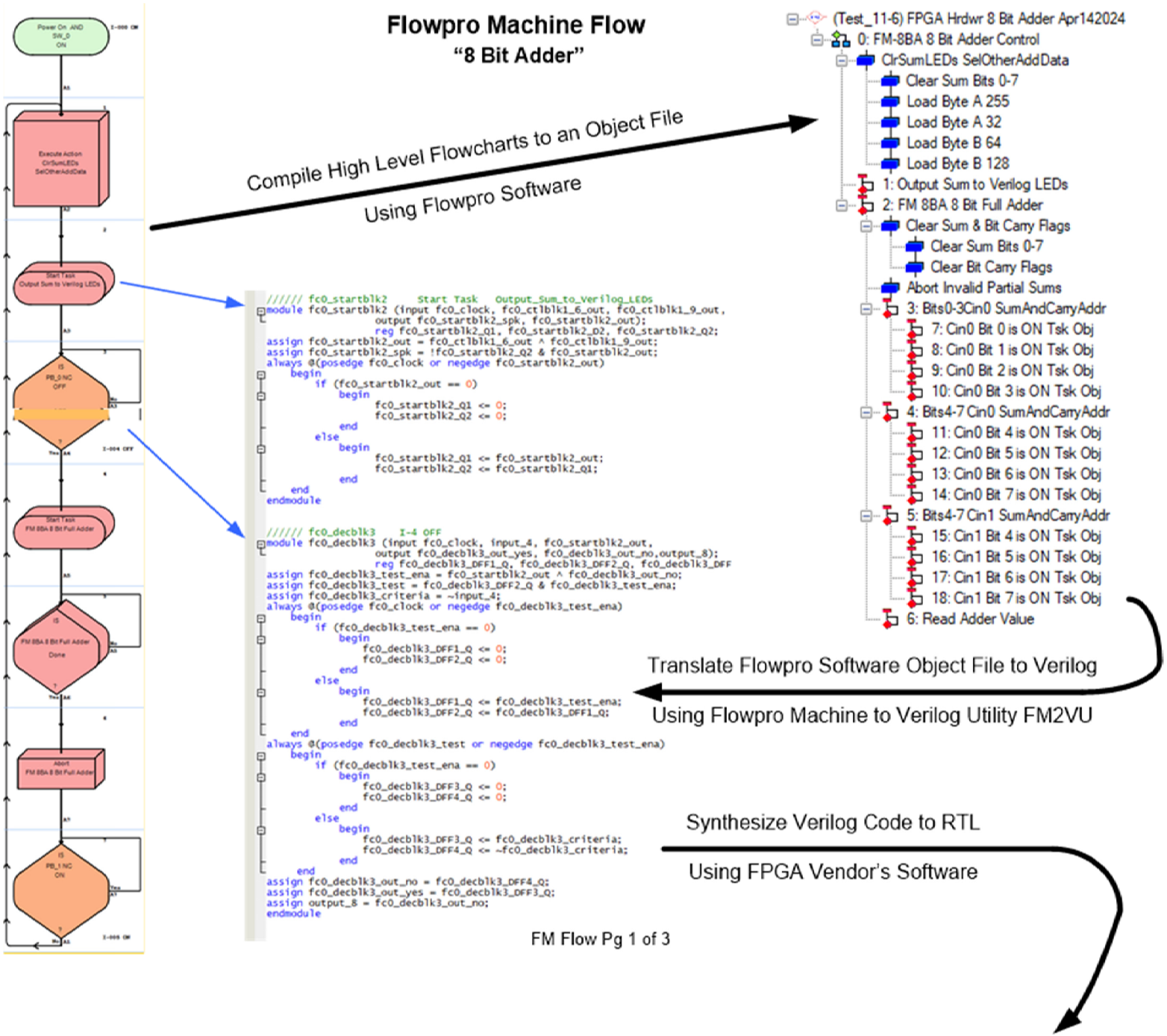
Flowpro Machine Flow
"8 Bit Adder"
Compile High Level Flowcharts to an Object File
Using Flowpro Software
(Test_11-6) FPGA Hrdwr 8 Bit Adder Apr142024
0: FM-8BA 8 Bit Adder Control
ClrSumLEDs SelOtherAddData
Clear Sum Bits 0-7
Load Byte A 255
Load Byte A 32
Load Byte B 64
Load Byte B 128
1: Output Sum to Verilog LEDs
2: FM 8BA 8 Bit Full Adder
Clear Sum & Bit Carry Flags
Clear Sum Bits 0-7
Clear Bit Carry Flags
Abort Invalid Partial Sums
3: Bits0-3Cin0 SumAndCarryAddr
7: Cin0 Bit 0 is ON Tsk Obj
8: Cin0 Bit 1 is ON Tsk Obj
9: Cin0 Bit 2 is ON Tsk Obj
10: Cin0 Bit 3 is ON Tsk Obj
4: Bits4-7 Cin0 SumAndCarryAddr
11: Cin0 Bit 4 is ON Tsk Obj
12: Cin0 Bit 5 is ON Tsk Obj
13: Cin0 Bit 6 is ON Tsk Obj
14: Cin0 Bit 7 is ON Tsk Obj
5: Bits4-7 Cin1 SumAndCarryAddr
15: Cin1 Bit 4 is ON Tsk Obj
16: Cin1 Bit 5 is ON Tsk Obj
17: Cin1 Bit 6 is ON Tsk Obj
18: Cin1 Bit 7 is ON Tsk Obj
6: Read Adder Value
Translate Flowpro Software Object File to Verilog
Using Flowpro Machine to Verilog Utility FM2VU
Synthesize Verilog Code to RTL
Using FPGA Vendor's Software
FM Flow Pg 1 of 3

Fig. 1

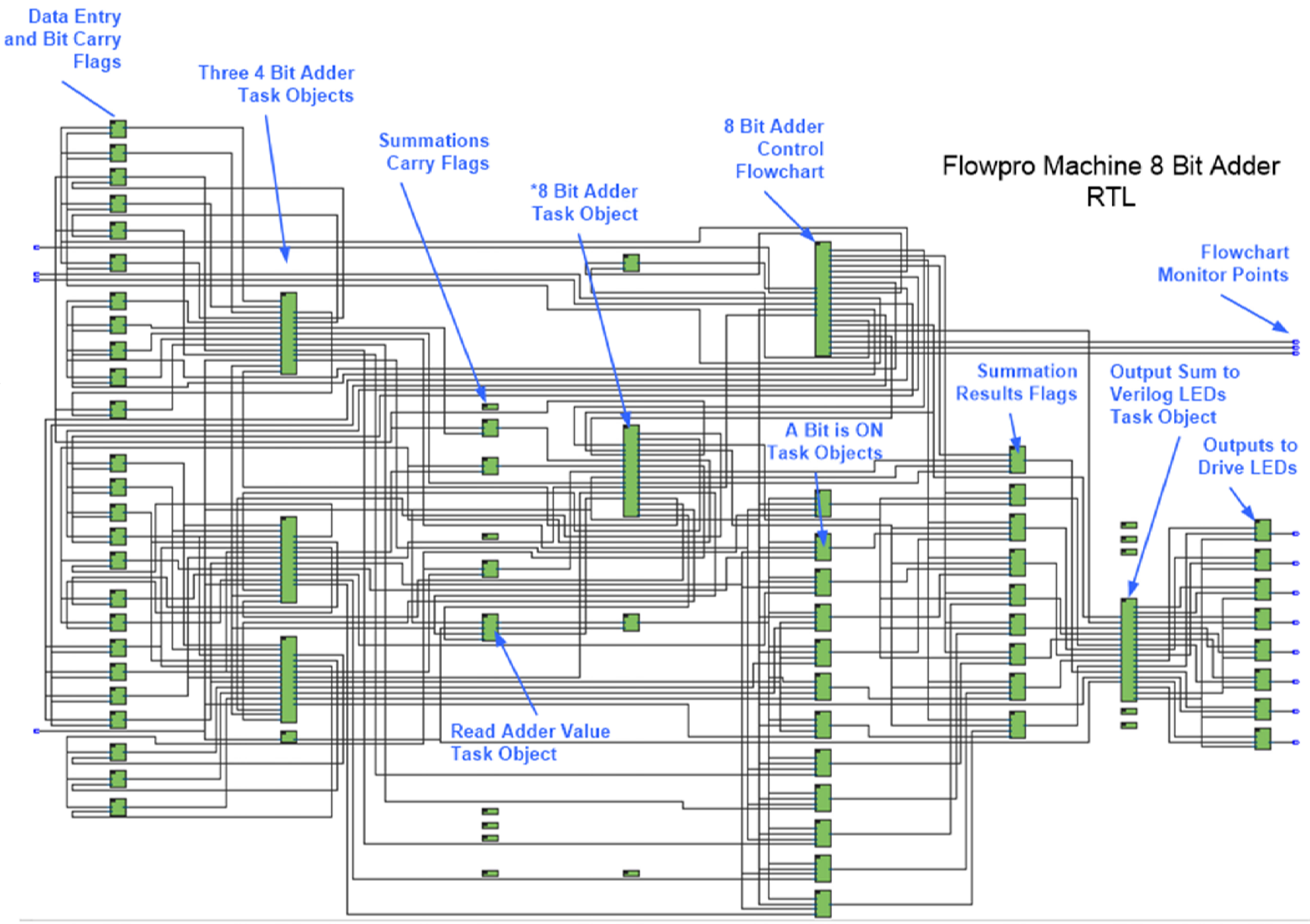
Data Entry and Bit Carry Flags
Three 4 Bit Adder Task Objects
Summations Carry Flags
*8 Bit Adder Task Object
8 Bit Adder Control Flowchart
Flowpro Machine 8 Bit Adder RTL
Flowchart Monitor Points
Summation Results Flags
Output Sum to Verilog LEDs Task Object
A Bit is ON Task Objects
Outputs to Drive LEDs
Read Adder Value Task Object

Fig. 2

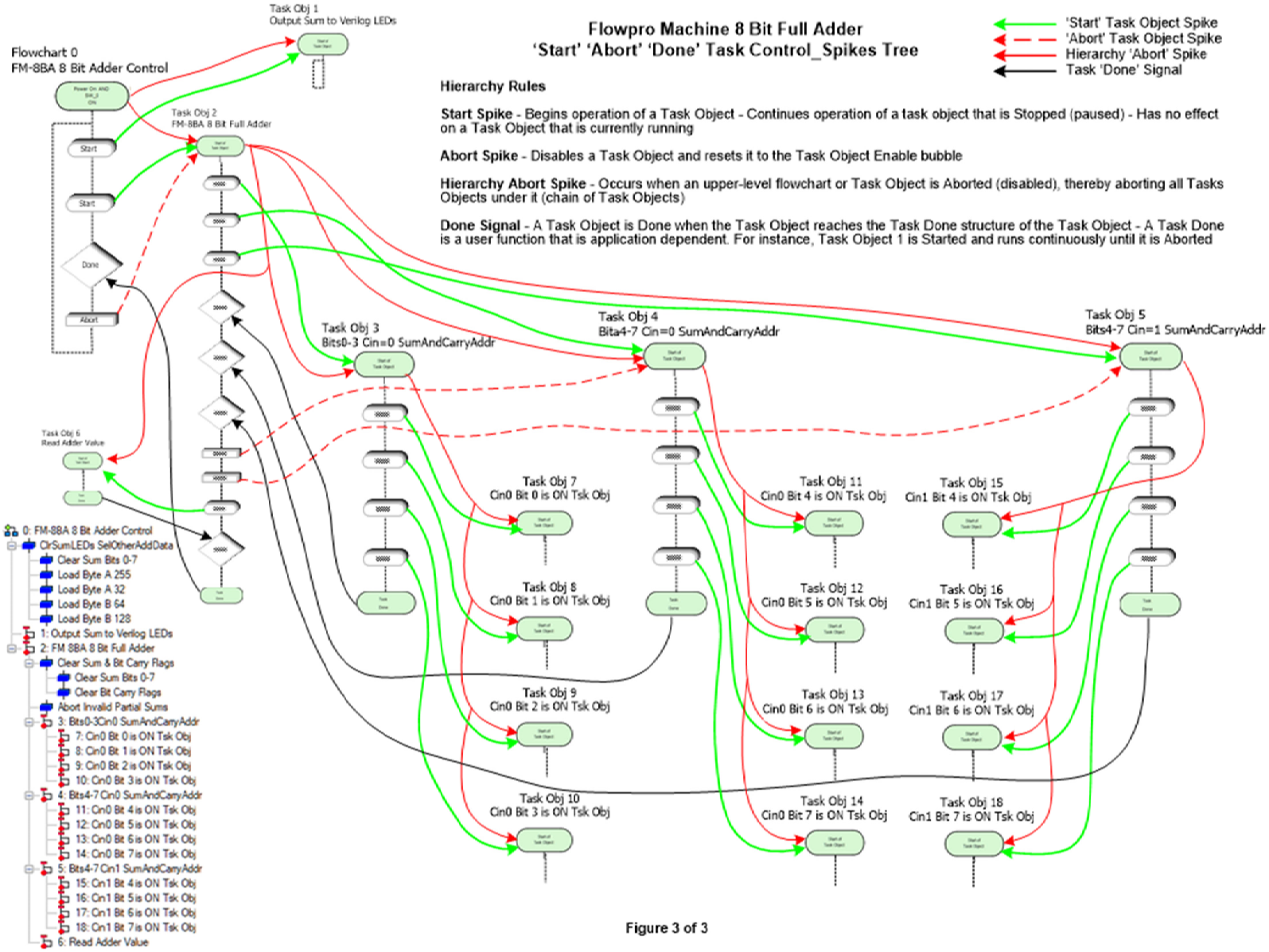

Figure 3 of 3

Fig. 3