The invention claimed is:

1. A parallel processing processor comprising: an algorithmic matching pipelined compiler; and a plurality of reusable algorithmic pipelined cores, each reusable algorithmic pipelined core comprising: a processing unit; and an array of reconfigurable, field programmable gates, wherein the field programmable gates are programmed by the algorithmic matching pipelined compiler, such that the algorithmic matching pipelined compiler is a precompiler that precompiles source code designed for operation on a standard processor without parallel processing for processing by the processing unit, wherein the precompiler is configured to precompile a standard higher level software language written not for the parallel processing processor but for a type of conventional non-reconfigurable processor, and the precompiler generates machine code for the reusable algorithmic pipelined cores by utilizing the type of conventional non-reconfigurable processor for which the standard higher level software language was written to generate machine code to configure the array of reconfigurable, field programmable gates, and the processing unit and the algorithmic matching pipelined compiler configures the field programmable gates to operate as pipelined, parallel processors, such that the source code designed for operation on a standard processor, without parallel processing, processes in parallel.

2. The parallel processing processor of claim 1, wherein the standard higher level software language is C or C++.

3. The parallel processing processor of claim 1, wherein the reusable algorithmic pipelined core comprises a pool of computers configured to process algorithms as needed for a particular calculation, based on output from the precompiler, and the field programmable gates are configured to complete tasks without any further overhead from the processing unit.

4. The parallel processing processor of claim 3, further comprising an intelligent bus controller or logical processor, wherein the intelligent bus controller or logical processor performs all of the logical functions processed by the processor.

5. The parallel processing processor of claim 3, further comprising a logical processor and a master bus switch, and the logical processor comprises reconfigurable logic functions for controlling of the master bus switch.

6. The parallel processing processor of claim 5, further comprising a digital signal processor, wherein the digital signal processor comprises a reconfigurable mathematical processor for performing mathematical calculations.

7. The parallel processing processor of claim 6, wherein the master bus switch is a matrix bus router or switch comprising a circuit reconfigurably programmable by the precompiler and the processing unit, such that data and results are routed from the reusable algorithmic pipelined core to another reusable algorithmic pipelined core to complete an algorithm, without any further intervention from a central or peripheral processor during processing of the algorithm, reducing overhead by pipelining compared to static, unreconfigurable hardware, which requires intervention by a central processor or peripheral processor to direct data and results in and out of arithmetic processing units.

8. The parallel processing processor of claim 7, wherein the logical processor processes logical decisions and iterative loops and result memory is provided by the logical processor for learning algorithms.

9. A parallel processing system comprising a parallel processing processor according to claim 4, comprising the steps of: processing all of the mathematical operations using digital signal processors of one or more of the plurality of reusable algorithmic pipelined cores; and processing all of the logic functions using the logical processors of the parallel processing processor.

10. The system of claim 9, further comprising the step of configuring the plurality of reusable algorithmic pipelined cores as a pool of processors, and the pool of processors are reconfigurable by programming, alone, without any change to the hardware.

11. The system of claim 10, wherein the step of configuring configures all of the plurality of

reusable algorithmic pipelined cores to process algorithms in parallel without any further intervention from a central or peripheral processor to direct data and results in and out of arithmetic processing units.

12. The system of claim 11, wherein the algorithmic matching pipelined compiler is a precompiler, and the logical processor of each of the plurality of reusable algorithmic pipelined cores uses memory blocks configured by the precompiler, as look up tables and registers for constants or learned values.

13. The system of claim 12, further comprising setting up the look up tables by the logic processor, wherein the look up tables are an n-bit look up table, and the n-bit look up table is used to encode an n-bit Boolean logic function as truth tables.

14. The system of claim 9, further comprising: generating machine code from a standard higher level software language written for a conventional, non-reconfigurable and non-pipelined, general purpose computer processor, using the algorithmic matching pipelined compiler for one or more of the plurality of reusable algorithmic pipelined cores.

15. The system of claim 14, wherein the standard higher level software language is written for a type of conventional, non-reconfigurable processor, and the step of generating machine code comprises the algorithmic matching pipelined compiler, as a precompiler, utilizing the type of conventional, non-reconfigurable processor to generate machine code for configuring the array of reconfigurable, field programmable gates of each of the plurality of reusable algorithmic pipelined cores.

16. The system of claim 15, wherein the system comprises at least one processor of the type of conventional, non-reconfigurable processor for which the standard higher level software language is written, and the at least one processor of the type of conventional, non-reconfigurable processor for which the standard higher level software language is written generates machine code for configuring the array of reconfigurable, field programmable gates of each of the plurality of reusable algorithmic pipelined cores.

17. The system of claim 16, wherein each of the plurality of reusable algorithmic pipelined cores is configured to independently solve complex mathematical and logic algorithms without further intervention by the at least one processor of the type of conventional, non-reconfigurable processor for which the standard higher level software language is written.

18. The system of claim 17, further comprising inputting values into the system, and the system outputs a solution to a master bus switch of the system, without further intervention.

19. The system of claim 18, wherein the plurality of reusable algorithmic pipelined cores comprises 2000 reusable algorithmic pipelined cores.

20. The system of claim 19, wherein the system operates 360 trillion instructions per second with a 500 MHz clock speed.

21. The system of claim 20, wherein the system has a latency for input of data, but pipelining reduces overhead such that the system is configured to execute a calculation and output a result on each clock from each reusable algorithmic pipelined core after the latency is initiated.