**On-line HW/SW partitioning and co-scheduling in reconfigurable computing systems**

* **Reconfigurable computing (RC) systems**

A reconfigurable computing system typically contains one or more processors and a reconfigurable fabric upon which custom functional units can be built.

Organization of RC systems with respect to the coupling of the RPU to the host computer is shown in Figure (1). The processor(s) executes sequential and non-critical code, HDL is mapped to reconfigurable fabric. Reconfigurable logic provides advantage of the parallelism. RCs based on Field Programmable Gate Arrays (FPGAs) are an attractive alternative. The resulting FPGA combines the best of both general purpose and custom IC. It is faster and smaller than general-purpose hardware, yet compared to an IC, it has smaller NRE costs and transition costs. FPGAs can be easily re-customized without modifying the hardware by designing and loading a different configuration. A reconfigurable computer could be upgraded, or even reconfigured for a completely different function, from a remote location.

Reconfigurable computing has several advantages and drawbacks. They are listed below.

**Advantages**

* **Greater Functionality**

It is possible to achieve greater functionality with a simpler hardware design. The required logic can be stored in memory and hence the cost of supporting additional features is reduced to the cost of the memory required to store the logic design. This is very much useful in mobile communication domain where protocol can be easily modified to newer protocol and stored in memory and then hardware can be reconfigured to achieve the required functionality. Compelling advantage includes increased speed, reduced energy and power consumption. A study reports that depending on the particular device used moving critical software loops to reconfigurable hardware results in average energy savings of 35% to 70% with an average speedup of 3 to 7 times. [4]

* **Embedded Characteristics**

In general-purpose computing processors common piece of silicon could be configured, after fabrication, to solve any computing task. This meant many applications could share commodity economics for the production of a single IC and the same IC could be used to solve different problems at different points in time. General-purpose computing meant engineers could *program* the component to do things which the original IC manufacturers never conceived. Embedded systems developers are much benefited from reconfigurable computing systems, especially with the introduction of soft cores which can contain one or more instruction processors. [4]

All of these "general-purpose" characteristics are shared by reconfigurable computing. Instead of computing a function by sequencing through a set of operations in *time* (like a processor), reconfigurable computers compute a function by configuring functional units and wiring them up in *space*. This allows parallel computation of specific, configured operations, like a custom ASIC. Also it can also be reconfigured. The reconfigurable hardware fabric can be easily and quickly modified from a remote location to upgrade its performance. It can be modified to perform a completely different function. Hence, non-recurring engineering (NRE) costs of reconfigurable computing are lower than that of a custom ASIC.

* **Lower System Cost**

By eliminating the ASIC design lower system cost on a low-volume product is achieved. For higher-volume products, the production cost of fixed hardware is actually very much lower. In the case of ASIC and general purpose hardware designs technical obsolescence drives up the cost of systems. Reconfigurable computing systems are upgradeable and extend the useful life of the system. This reduces lifetime costs.

* **Reduced Time to Market**

Reduced time-to-market is the final advantage of reconfigurable computing. Since ASIC is no longer used in reconfigurable computing large amount of development effort is reduced. The logic design remains flexible even after the product is shipped. Design can be sent to market with minimum requirements and later additional features can be added without any change in physical device (or system). Thus reconfigurable computing allows incremental design flow.

These advantages lead reconfigurable computers to serve as powerful tools for many applications. The applications include research and development tools for sophisticated electronic systems such as ASICs and printed circuit boards (PCBs). Simulation tools for these systems do not exist. Also prototype fabrication is expensive and time consuming. A reconfigurable computer can serve as an affordable, fast, and accurate tool for verifying electronic designs

**Disadvantages**

Two severe disadvantages of reconfigurable computing can be observed. They are the time that the chip takes to reconfigure itself to a given task, and the difficulty in programming such chips. Dynamic reconfigurable computing has several different complex issues. They are design space, placement, routing, timing, consistency and development tools. Each of these is discussed below.

* **Placement Issues**

In order to reconfigure a new hardware, it requires having ample space to place the new hardware. The component placement issue becomes complex if the component needs to be placed near special resources like built- in memory, I/O pins or DLLs on the FPGA.

* **Routing Issues**

Existing components has to be connected to the components newly reconfigured. The ports must be available to interface new components. The same ports must have also been used under the oldconfiguration. To accomplish this orientation of the components should be in a workable fashion.

* **Timing Issues**

Newly configured hardware must meet the timing requirement for the efficient operation of the circuit. Longer wires between components may affect the timing. Optimal speed should be attainable after dynamically reconfiguring the device. Over timing or under timing the new added design may yield erroneous result.

* **Consistency Issues**

Static or dynamic reconfiguration of the device should not degrade computational consistency of the design. This issue becomes critical when the FPGA is partially reconfigured and interfaced with existing design. Adding new components to the device by reconfigurable fabric should not erase or alter the existing design in the device. (Or memory). There should be some safe methods to store the bit stream to the memory.

* **Development Tools**

Commercial development tools for dynamic reconfigurable computing are still under development stage. The lack of commercially available tools for the specification to implementation stages of the digital design is still a bottleneck. The available tools require enormous human intervention to implement the complete system.

Reconfigurable computing has been an emerging technology, but only recently has the possibility of developing such systems for space become exploited. The idea of reconfigurable computing in space is a hardware designer’s dream since the notion of fixing a hardware bug or updating a hardware-implemented algorithm has never been available. Space missions are critical by nature and, therefore, having the option of reconfiguring a system once launched provides numerous advantages and a great deal of flexibility.

Reconfigurable computing technology is the ability to modify a computer system's hardware architecture in real time. Although originally proposed in the late 1960s by a researcher at UCLA, reconfigurable computing is a relatively new field of study. The decades long delay had mostly to do with a lack of acceptable reconfigurable hardware. Interest in this field was first triggered- off late in 2002 when a small Silicon Valley start up called Quick Silver Technologies announced what it called the Adaptive Computing Machine(ACM), a new class of digital integrated circuit that can be embedded directly into a mobile device and will enable hardware to be programmed almost as if it were a piece of software, For example, take 3 common applications that the average mobile phone performs seamlessly: search for a local cellphone; verify whether the number represents an authorized user then make the connection. Today the 3 operations are performed by 3 different chips inside the handset. With the new adaptive technology, a single chip can be reconfigured by a software instruction to assume different hardware functions and to perform all 3 applications one after the another   
  
The earliest reconfigurable computing systems predate even digital computers. Before digital logic scientific and engineering computations were done on programmable analog computers: big banks of op amps, comparators, multipliers and passive components interconnected via a plug board and patch cords. By connecting components together, the very clever user could implement a network whose node obeyed a set of differential equation solver, capable of deployment- time reconfigurability. Toward the end of its era, the analog computer was combined with relay banks, and later with digital computers, to form hybrids. These machines could reconfigure themselves between execution sequences, providing an early form of yet another category of configurability. Some hybrid computer programmers become experts at juggling configurations while holding data in sample and to extend the range of these systems  
  
The first moves toward really fluid reconfigurability came with the advent of embeddable digital computers. With the characteristics of a system defined by software in RAM, nothing could be simpler. Changing the operation of the system at installation, in response to changing data or even on the fly, is the matter of loading a different application. Variants on this theme included tightly coupled networks of computers in which the network topology could adapt to changing data flows, and even computers that could change their instruction sets in response to changing application demands.  
  
But the first explorations into what most people today mean by the term reconfigurable computing came after the development of large SRAM- based FPGAs. The devices provided a fabric of logic cells and interconnects that could be altered- albeit with some difficulty - to create just about any logic netlist that would fit into the chip. Researches quickly seized upon the parts and began experimenting with deployment tie reconfiguration creating a hardwired digital network designed for a specific algorithm.  
  
Experiments with reconfigurability in FPGAs identified two promising advantages: reduction of size or power consumption of the hardware, and increases in performance. Often the two types of advantages came together, rather than separately. The advantages, it turned out, came with only few quite specific techniques. One of these was simple: reuse of hardware. If it is organize a system in such a way that it has several distinct, non overlappimg operating modes, then you can save hardware by configuring a programmable fabric to execute in one mode, stopping, then configuring it to operate in another mode  
  
A number of companies are currently working in this area. Most of the big players in the conventional DSP/ASIC area -Texas ,IBM, Motorola, Intel- are known to be working overtime to come up with reconfigurable designs of their own.

**RECONFIGURABLE COMPUTING SYSTEMS**  
  
Current computers are fixed hardware systems based upon microprocessors. As powerful as the microprocessor is, it must handle far more functions than just the application at hand. With each new generation of microprocessors, the applications performance increases only incrementally.In many cases the application must be rewritten to achieve this incremental performance enhancement. Traditional fixed hardware may be classified into three categories: Logic(Gate Arrays, PALS etc.), Embedded control (controllers eg ASICs & Custom VLSI Devices) and Computers(Microprocessors eg x 86, 68000, Power PC).

Reconfigurable Computing Systems are those computing platforms whose architecture can be modified by the software to suit the application at hand. To get the maximum through put, an algorithm must be placed in hardware ( eg. ASIC, DSP, etc) Dramatic performance gains are obtained through the 'hardwiring' of the algorithm. In a recofigurable computing system, the "Hardwaring takes place on a function by function basis as the application executes

Reconfigurable computing systems allow executing tasks in a true multitasking manner. Such systems share the reconfigurable device and processing unit as computing resources which leads to highly dynamic allocation situations. To manage such systems at runtime, a reconfigurable operating system is needed. The main part of this operating system is resource management unit which performs HW/SW partitioning, co-scheduling and placement of hardware tasks at run-time. In this paper, we present a heuristic for on-line integrated HW/SW partitioning and co-scheduling. We focus on on-line, non real-time and non-preemptive systems. The main characteristic of our heuristic is strong nexus between partitioning, scheduling and placement. Our heuristic prioritizes the arrived tasks according to different important parameters and partitions the sorted tasks according to their earliest finish time (EFT) on software and hardware processing units. A large variety of experiments have been conducted on the proposed algorithm using synthetic tasks. Obtained results show considerable benefits of this algorithm.

# Real-time multiple face detection and tracking

In recent years, processing the images that contain human faces has been a growing research interest because of establishment and development of automatic methods especially in security applications, compression, and perceptual user interface. In this paper, a new method has been proposed for multiple face detection and tracking in video frames. The proposed method uses skin color, edge and shape information, face detection, and dynamic movement analysis of faces for more accurate real-time multiple face detection and tracking purposes. One of the main advantages of the proposed method is its robustness against usual challenges in face tracking such as scaling, rotation, scene changes, fast movements, and partial occlusions.

# Declarative Testing: A Paradigm for Testing Software Applications

Traditional techniques to test a software application through the application's graphical user interface have a number of weaknesses. Manual testing is slow, expensive, and does not scale well as the size and complexity of the application increases. Software test automation which exercises an application through the application's UI using an API set can be difficult to maintain. We propose a software testing paradigm called declarative testing. In declarative testing, a test scenario focuses on what to accomplish rather than on the imperative details of how to manipulate the state of an application under test and verify the final application state against an expected state. Declarative testing is a test design paradigm which separates test automation code into conceptual Answer, Executor, and Verifier entities. Preliminary experience with declarative testing suggests that the modular characteristics of the paradigm may significantly enhance the ability of a testing effort to keep pace with the evolution of a software application during the application's development process.

# Classification Algorithms of Trojan Horse Detection Based on Behavior

Current anti-Trojan is almost signature-based strategies, which cannot detect new one. Behavior analysis, with the ability to detect Trojans with unknown signatures, is a technique of initiative defense. However, current behavior analysis based anti-Trojan strategies have the following problems: high false or failure alarm rate, poor efficiency, and poor user-friendly interface design, etc. The paper works on the design of an anti-Trojan oriented algorithm based on behavior analysis. And we construct a standard of anti-Trojan algorithm system and point the up-limit of the precision. We propose an improved hierarchical fuzzy classification algorithm which is specifically designed for anti-Trojan. Finally, we organize the experiment to get the results. The results show high classification accuracy using our algorithm. Compared to Bayesian algorithm, our algorithm have better performance.