

# The Coming Era of Field Programmability (or ASICs are Dead!)

## DesignCon 2001

**Chuck Fox**  
**President & CEO**  
**Chameleon Systems**  
**408 240-3300**  
[Chuck@cmln.com](mailto:Chuck@cmln.com)

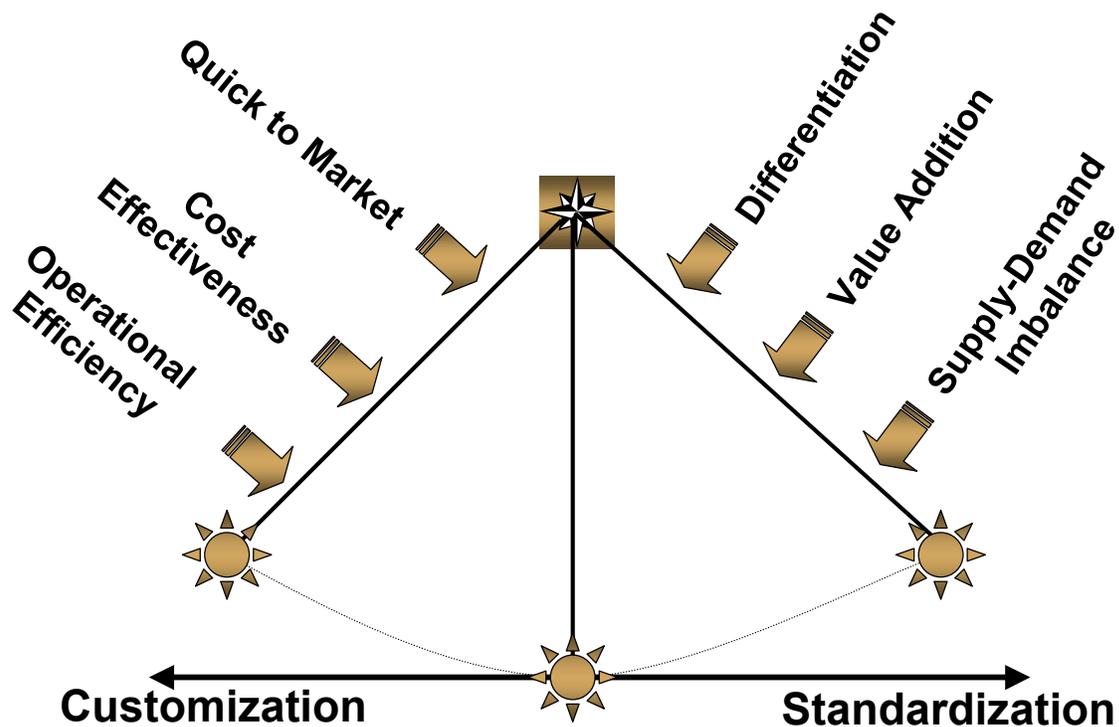
The title of my talk today is “The Coming Era of Field Programmability or ASICs are Dead”.

In my view, ASICs will continue to be an important design technology for a smaller and smaller set of high performance applications. In the communications field, ASICs will continue to be the design technology of choice when the problem is closer to the core of the network. However, as the problem moves closer to the access side programmable technologies, whether they be programmable standard products or field programmable chips, will and are becoming a desirable choice. The need for absolute performance in this segment gives way to the need for design flexibility, multiprotocol functionality and fast time-to-market. The trend shows no signs of abating.

ASICs began a slow death several years ago when FPGA design starts exceeded ASIC design starts. In my view, ASICs will become an increasingly important technology for a decreasing number of applications until they can no longer be an economically viable or “time-to-market viable” solution. They will become an important segment for exotic high performance design but not for the mainstream user. Some of you may be old enough to remember the naysayers who said that CMOS technology could never be fast enough to become a mainstream technology, that it would be relegated to low power applications only, and that bipolar technologies were to remain the dominant technology because of speed. Bipolar ASIC technology still exists for exotic high performance design but, of course, the investment behind CMOS technology R&D ensured that CMOS would be dominant. In my view, the same scenario will play out in the chip design area in the next 5-10 years. The investment in programmable technologies is exploding. This includes both programmable standard products (like network and communication processors) and field programmable products (like FPGAs). These two segments of programmable products are beginning to blur. The investment in ASIC technologies pales in comparison.

I’ll begin with a discussion of the design tradeoffs between performance and flexibility, between standardization and customization, I’ll describe the variety of software defined design platform trends, introducing the doctrine of reconfigurability, and end with some concrete examples of reconfigurable applications.

# Semiconductor Pendulum



Source: Dr. Tsugio Makimoto, Sony – FPL August 2000

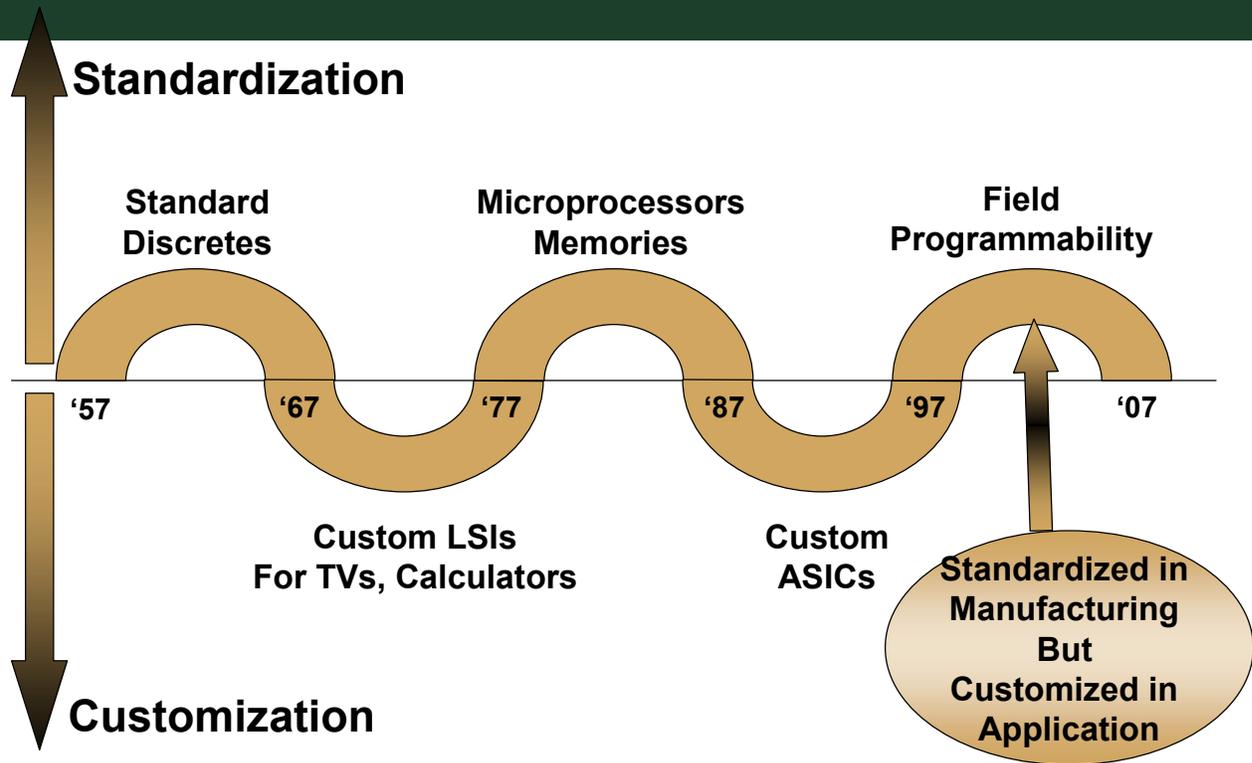
DESIGNCON 2001

CFox Chameleon Systems 2 0101

When I talk to customers what I hear is a need to increase performance, reduce chip cost, reduce power and differentiate. This says, “customize”. But I also hear them say they can’t get their ASICs done fast enough, they can’t find enough ASIC engineers, and they can’t manage their ASIC inventory. This screams “standard product”. The need for customization and standardization has never been greater.

Dr. Tsugio Makimoto from Sony Corporation summarized this nicely at field programmable logic conference in Austria last year. He described the opposing forces of customization and standardization. The need for quick time-to-market and cost effectiveness push the pendulum away from custom products and the need for differentiation and value addition push the pendulum away from standard products. This is the conundrum that many designers face.

# The Rising Wave of Field Programmability Makimoto's Wave



Source: Dr. Tsugio Makimoto, Sony – FPL August 2000

CFox Chameleon Systems 3 0101

DESIGNCON 2001

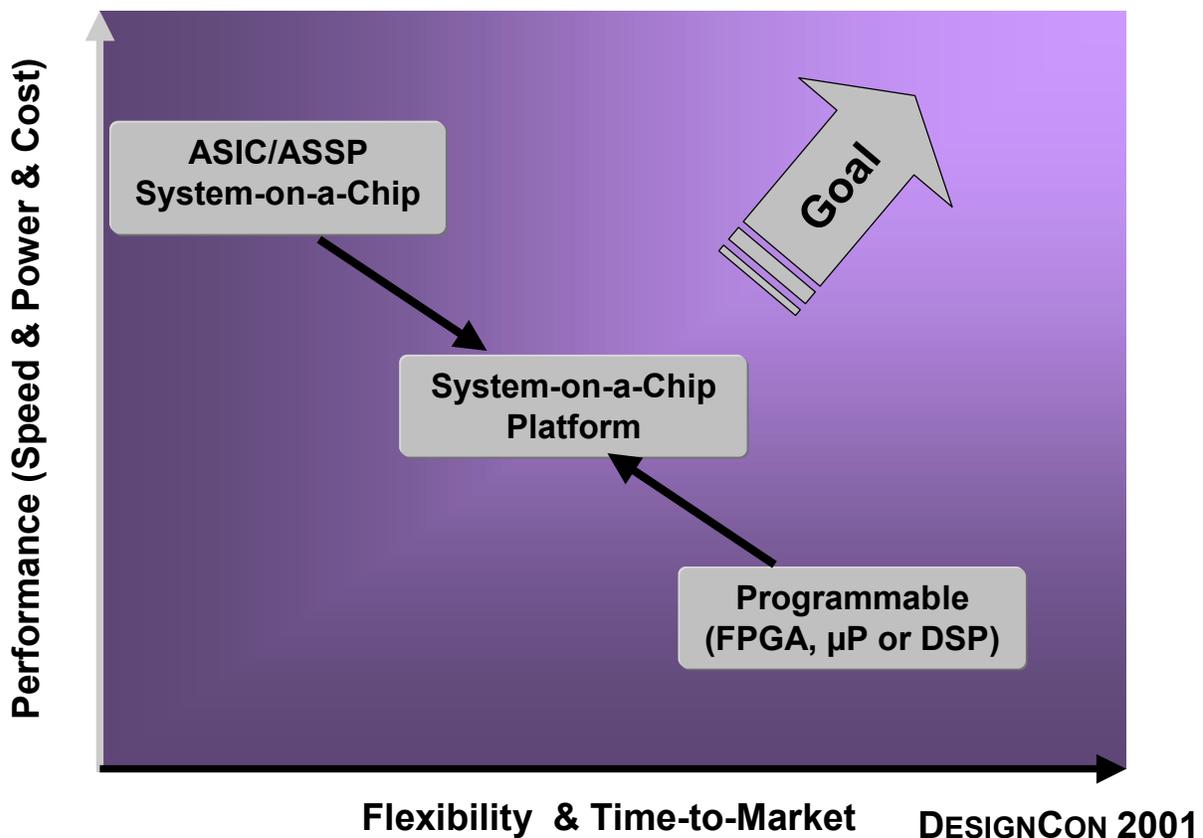
Dr. Makimoto also observed that there have been alternating cycles of standardization and customization in the semiconductor industry dating back to its beginning. The decade of 1957-1967 saw the take-off of the semiconductor industry based on discrete devices. Because these devices were interchangeable, a cycle of standardization began. The IC was invented in 1957 but the actual takeoff of the industry began around 1967. ICs were customized for specific applications such as TVs and calculators, so a cycle of customization began. As the calculator war escalated, the life cycles of calculator chips began shorter, production volumes decreased, resulting in decreased operational efficiency. That was the end of that customization cycle.

Intel invented the Microprocessor and first static memory ICs around 1971. The processor made it possible to provide different types of calculator functions by means of a program stored in memory, allowing a standard product to serve many applications. The microprocessor market reached \$1B in 1977 beginning the another cycle of standard products.

Thanks to the development of electronic design automation technology the application specific products became more feasible in the late 1970's with the introduction of the metal programmed gate array. In the early 80's LSI Logic pioneered the mass-market approach to ASIC design. The ASIC market surpassed the market for standard logic products in 1987, indicating the beginning of another cycle of customization.

The FPGA was invented in the mid 80's but only in the mid to late 90's did it surpass ASIC technologies in terms of design starts, and gate arrays in terms of market size. Dr. Makimoto predicted 10 years ago the field programmability would become a leading technology. That prediction has come true and we are now in a rapidly expanding period of field programmable standard products, that is, those products are standardized in manufacturing but customized in application.

# Design Tradeoffs



CFox Chameleon Systems 4 0101

So how what options do designers have to achieve both performance and time-to-market. When highest performance is the requirement, then ASIC System-on-a-Chip customized products or standard products (ASSPs) are the choice. Of course, they suffer from poor design flexibility and poor time-to-market. When best flexibility and best time-to-market are desired then of course programmable technologies are chosen.  $\mu$ Ps and DSPs are relatively slow and FPGAs are relatively expensive and power hungry.

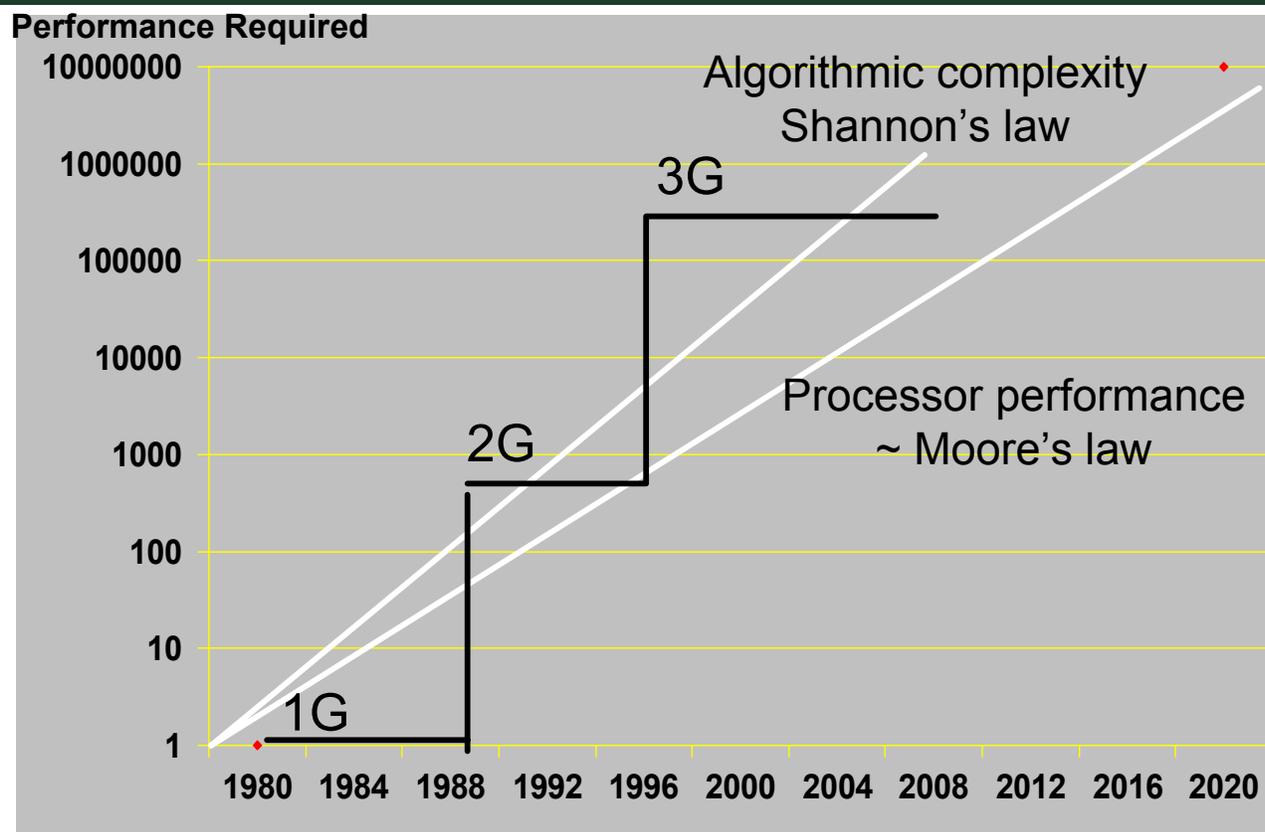
Recently a variety of ASIC and FPGA System-on-a-Chip Pre-defined Platform technologies have emerged in order to strike a better balance between performance and time-to-market.

What really happens, though, is that if the performance requirement can't be met then it doesn't matter how flexible it is or how fast you get it to market. The goal, of course, is to move up and to the right. That is,

- \* to reduce time-to-market or increase flexibility for a given performance
- \* And increase performance for a given time-to-market.

The easiest approach is to wait for Moore's law to catch up. However, in many markets the performance requirement is far outstripping the ability of Moore's law to keep up. Let me discuss one example I'm familiar with.

# Shannon beats Moore's law!



Source: Jan Rabaey, Berkeley Wireless Research Center

DESIGNCON 2001

CFox Chameleon Systems 5 0101

This chart plots performance requirements in wireless systems versus time. Moore's law, which says that processor performance doubles every 18 months, is plotted against Shannon's law which is a measure of how algorithm complexity is accelerating in the wireless world. What it says is that if you wait for your uP or DSP to get faster you won't meet the requirements for 2G and 3G wireless systems. You'll be short about an order of magnitude and that gap is increasing. You must take another approach to increase performance. You've got to do more processing on a single clock cycle to increase overall chip performance. Some type of parallelization of algorithms in an ASIC is typically the solution. However, just raw ASIC performance won't necessarily do it.

# A Plethora of Wireless Standards

Table 6-6 2G and 3G Modulation Parameters

Parameter Standard ⇒ ↓	Bandwidth Frame Size	Modulation Format	User Bitrate Kbps	Symbol (ksps), Bit rate (kbps) or Walsh codes	Chip Rate	Processing Gain
<b>2<sup>ND</sup> GENERATION STANDARDS (2G)</b>						
TDMA Cellular/PCS North America	30 kHz 20 msec	TDMA π/4-DQPSK	16.2	24.3 k sym/sec 48.6 kbps	NA	NA
CDMA Cellular North America	1.25 MHz 20 Msec	CDMA BPSK & QPSK	9.6 14.4	19.2 ksps per code 64 walsh codes	1.2288 Mcps	~ 20 dB
Personal Digital Cell. (PDC) Japan	25 kHz 20 msec	TDMA π/4-DQPSK	14	21 sym/sec 42 kbps	NA	NA
GSM 900, 1800, & 1900	200 kHz 4.615 msec	GSM .3 GMSK	22.8	270.833 ksym/sec 270.833 kbps	NA	NA
<b>ITU GLOBAL 3<sup>RD</sup> GENERATION STANDARDS (3G)</b>						
UWC-136 / EDGE (SC)	30 kHz 20 msec	TDMA 8-PSK	16.2	24.3 k sym/sec 72.9 kbps	NA	NA
UWC-136 / GSM / (SC) Outdoor	200 kHz 4.615 msec	GMSK 8-PSK	22.8	270.833 ksym/sec 812.499 kbps	NA	NA
UWC-136 HS Indoor (ITU-SC)	1.6 MHz 4.615 msec	B-O-QAM Q-O-QAM	> 2 Mbps	2.6 Mcps > 2.6 Mbps	NA	NA
CDMA-2000 (MC) (1RXX) (1 carrier)	1.25 MHz 20 Msec	CDMA QPSK	9.6 - 14.4 > 384	19.2 ksps per code 128 walsh codes	1.2288 Mcps	~ 20 dB
CDMA-2000 (MC) (3RXX) (3 carriers)	1.25 MHz 20 Msec	CDMA QPSK	9.6 - 14.4 > 384 >2000	19.2 ksps per code 128 walsh codes	1.2288 Mcps	~ 20 dB
WCDMA (DS) (Harmonization still in progress)	5 MHz 10 msec	CDMA BPSK QPSK	9.6 - 14.4 > 384 >2000	Variable ~ 15 – 960 ksps 4 – 256 Walsh Code	3.84 Mcps	Variable ~ 6 – 24 dB

I apologize for this eye chart. When you take a look at the complexity of mobile wireless standards, the need for some type of software programmability becomes obvious. There are ten different 2G and 3G standards at this time. You can see the same situation in many other areas such as the fixed broadband wireless market, the wireless LAN market and the xDSL market.

# ASIC Fundamentals Deteriorating

- Time to Market Challenges
  - Design & Tool Complexity
  - \$1M mask sets
  - Scarcity of Good ASIC Engineers
  - Design Changes
  
- Development Cost Challenges
  - Design & Tool Complexity
  - \$1M mask sets
  - Scarcity of Good ASIC Engineers
  - Design Changes

The fundamentals of ASIC design are deteriorating. Design complexity is increasing, tool complexity is increasing, the era of \$1M mask sets is in sight, you can't find enough good ASICs engineers to get designs done and when you do, the marketing guys change the design requirements and you restart. All of these factors affect both the time-to-market and development costs of ASIC design.

So do you know what you're biggest enemy is?

# THE ENEMY

# TIME

CFox Chameleon Systems 8 0101

**DESIGNCON 2001**

Today, time is your enemy!

A massive shift is taking place in chip design today. The future belongs to those who can exploit the benefits of software programmability.

If you think that ASICs are the only way to solve your design problem, TIME will leave you behind.

If you think software & field programmable products are always expensive and slow, TIME will leave you behind.

If you think the silicon inefficiency of software & field programmable products makes them niche or prototyping products only, TIME will leave you behind. If you think “real men only design ASICs”, TIME will leave you behind.

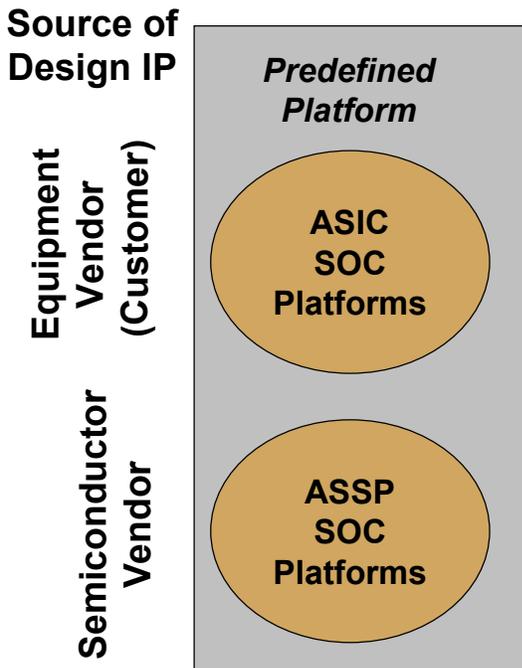
Do we have any chip designers in the audience? Great. When you design any chip you'd like to have zero cost, zero power, zero delay and do it in zero time. Typically, you'll tradeoff cost/power/delay to meet the chip specification and then work like hell to figure out a way to reduce design time using a variety of EDA, verification, pre-defined SOC platforms and/or FPGA approaches. At worst it takes well more than a year, at best it takes many months. In any event, it almost always takes too long. You are always under pressure to reduce this time-to-market. You can't go anywhere without hearing about the need to reduce design time. We have a crisis in design time.

The new chip innovators are designing software programmable chip platforms to reduce:

- The Time-to-Design Change
- The Time-to-New Feature
- The Time-to-New Protocol
- The Time-to-New Standard
- The Time-to-New Algorithm

Lets look at several different platform approaches all designed to reduce time-to-something new....

# Software Defined Platforms



CFox Chameleon Systems 9 0101

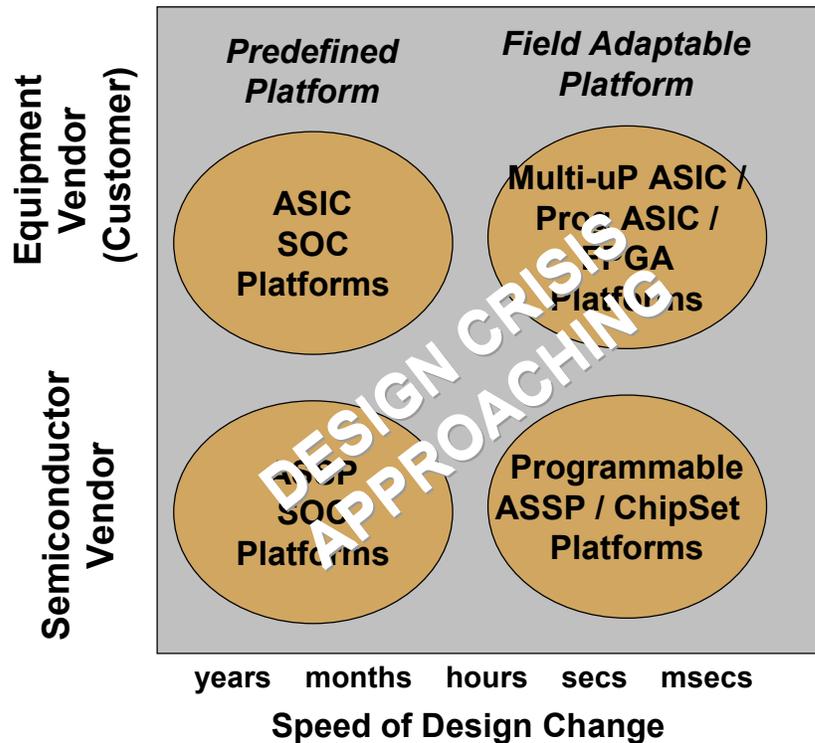
DESIGNCON 2001

The first approach is what I call the Pre-Defined Platform Approach. For ASIC vendors, this consists of the pre-defined SOC platform in which the ASIC vendor pre-defines and pre-designs a set of IP blocks together to reduce the time-to-design for the equipment vendor's engineers. But the resulting design is still an ASIC and still suffers from the time-to-market, inflexibility and development costs issues of a custom ASIC design.

The other case is the standard product (ASSP) created by the semiconductor vendor. Innovative approaches to creating internal SOC platforms are abundant in the semiconductor industry. These internal SOC platforms are typically a single standard product that is modified slightly to create multiple products, with pre-defined IP blocks as well. Sometimes, it's just a bond-out option, other times it's the same die with some features "removed" by not testing them but shipping the same actual die and marking it with the new product name. In all of cases you just build in all of the different functions/protocols/standards/algorithms on one ASIC or ASSP chip and just then design logic to switch between them. This can be expensive especially if the alternative functions or algorithms aren't used much. One vendor estimated that the number of transistors actually active at any one time in a typical ASIC or ASSP approach was between 10 and 30%. But it's cheaper than taping out a new chip.

# Software Defined Platforms

## Source of Design IP



CFox Chameleon Systems 10 0101

**DESIGNCON 2001**

The 2nd approach is what I'll call the "field adaptable approach". In this case the objective is to design in some type of end-user controllable field programmability. Building field adaptable chips allows them to be reprogrammed for different functions through software downloads, essentially software drivers, by updating logic cells, configuring programmable interconnect, loading registers to steer functionality or programming on-board embedded processors of some type. The multi-processor ASIC is becoming popular with the availability of cheap, easy to integrate embedded processor IP cores. A second method is sometimes known as the "Programmable ASIC", which sounds like an oxymoron to me. In this case, a fixed function ASIC is designed with pockets of embedded processors or just register files for some software programmability. In the end of course, these chips help the time-to-market issue but they are still ASICs and suffer from the fundamental ASIC deteriorating issues. The FPGA approach is well understood in its ability to update functionality in the field but still suffers from cost and power issues.

In the case where the total functionality of the chip is defined by the semiconductor vendor we would call this a programmable standard product or programmable ASSP or programmable chip-set. The variety of network processors usually falls into this category.

In all of these cases, the amount of software programming is usually limited to a very small % of the chip, even in the FPGA case. These approaches are great to reduce time-to-market for new functions but they usually come at a performance, power, cost or flexibility penalty which often cannot be tolerated.

The pre-defined platform and field adaptable platforms are steps in the right direction. However, they alone will not meet the performance and time-to-market challenges coming this decade. They will not satisfy the conflicting demands of standard products and customization. They will not effectively address the exploding development costs of deep-submicron design and the scarcity of top ASIC engineers. They still make huge tradeoffs between performance and flexibility. We have a design crisis heading our way. Another approach is needed. Another approach is coming.

# THE DOCTRINE OF RECONFIGURABILITY

## *WASTE TRANSISTORS THEN REUSE THEM*

CFox Chameleon Systems 11 0101

**DESIGNCON 2001**

The new approach is the “chip on demand” approach based on reconfigurable technology. This essence of this approach is captured by what I’ll call “the doctrine of reconfigurability”.

Waste transistors and then reuse them.

So you’re sitting there thinking, what is this guy talking about?

First of all, we should waste transistors. We should turn the conventional design wisdom of minimizing transistors or minimizing die size upside down. Transistors are getting cheaper and smaller. If we use transistors not only to create more on-chip fixed functionality or more MIPS but to use transistors to create the architecture where the chip is defined by “programming bits”, literally 1’s and 0’s, we can define an entire chip by a small file of bits. In this case chips can actually be created by simply loaded the bit file into the chip from RAM on the board, over the Internet or through a wireless connection. One larger chip can effectively represent 10’s or 100’s or an infinite number of chips by simply reprogramming the chip.

Secondly, we should reuse the transistors many times. If the bits can be loaded in real-time (microseconds or less) then the chip can dynamically change its function in response to the functional or algorithmic requirements of the data stream being presented to the chip. This takes field programmability to a whole new level. This is the vision of reconfigurable computing

“Much in the same way that the microprocessor has reshaped the last three decades,

*we believe that **reconfigurable computing technology** will be a key trend reshaping the semiconductor industry in the twenty-first century.*

The traditional distinction between what is hardware and what is software is blurring, and we will see leading-edge companies that are able to capitalize on this paradigm shift”

Source: (6/99) **H&Q**

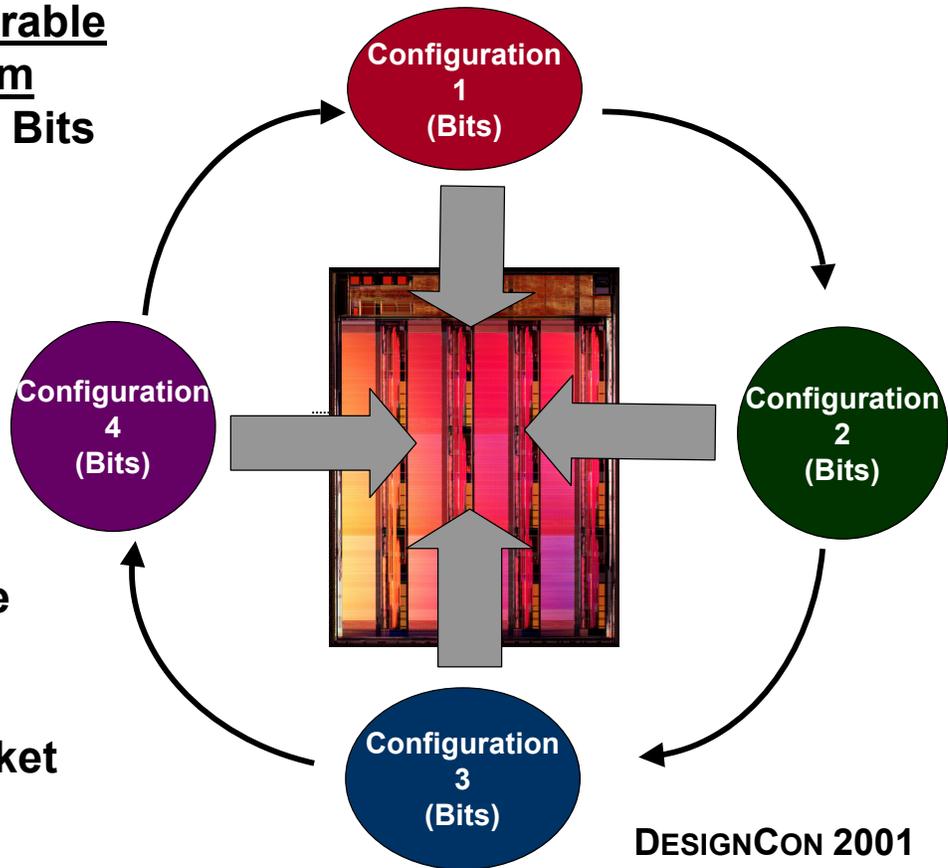
# “Chip on Demand” Approach

**Reconfigurable  
Platform**  
Functions are Bits

Program both  
functions  
AND  
interconnect

## **Benefits**

- Performance
- Power
- Flexibility
- Time-to-Market



**DESIGNCON 2001**

CFox Chameleon Systems 13 0101

In this case the entire chip and all of its functions are simply defined by a series of bits. The bits are created in advance by a design system and then loaded into the chip as needed. These bits describe both the processing functions required AND the interconnect of those functions. The bits are loaded in microseconds and therefore the entire chip architecture is defined and redefined by the application in real-time. The chip architecture can be optimized to match the function and algorithm requirement instead of the function or algorithm being changed to match a fixed architecture. The resulting benefit is a reduction in power consumption, an increase in system performance and an effective reduction in cost/delivered function while providing a remarkable amount of design flexibility, and an incredibly fast time-to-market. Adding new functions in the “system” become free once the fixed price of reconfigurability is paid, that is the overhead of additional transistors. This is as close as we get to creating chips in ZERO time.

# “Chip on Demand” Approach

## The Network is The Architecture

CFox Chameleon Systems 14 0101

**DESIGNCON 2001**

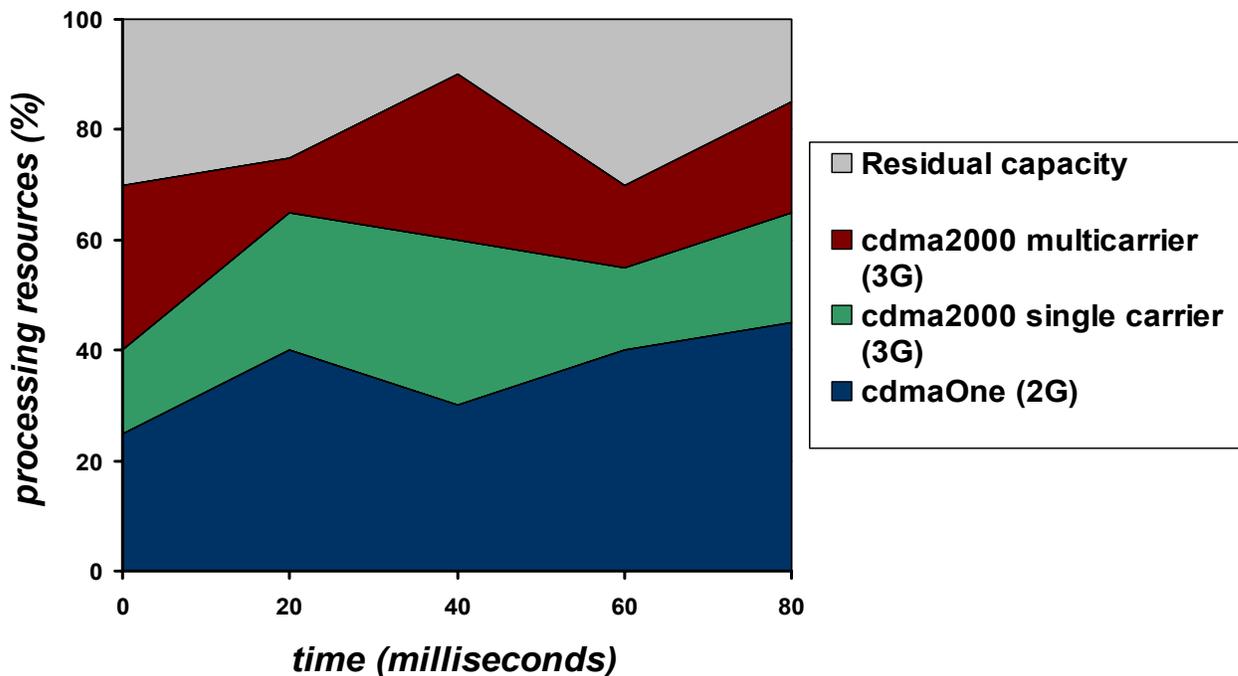
As I said, the interconnect programming is the key definition of reconfigurability. Programmability differentiates this approach from the traditional multi-processor or programmable standard product approaches. In all of those cases even though the programmability of the chip is limited to functions or processors. Software programming of the interconnect gives the designer the ability to create an architecture with whatever dataflow, parallelization, or pipelining architecture needed. Effectively the designer creates and customizes the on-chip “network of processing elements”. In this case the “network is the architecture”.

I see a future in enabling equipment providers software designers to create their own chips and their own architectures simply by defined a set of “bits” and programming a pre-defined reconfigurable platform. In this case, time is your friend and the system architecture is designed with time multiplexing of functions as a cornerstone of the architecture. Interesting that in this approach between 60 and 90% of the transistors in a chip are being used at any one time because the architecture is optimized for the algorithm and redundant less often used functions don’t need to be fixed in hardware. In non-reconfigurable approaches 70-90% of the transistors are not doing anything most of the time.

Mark this prediction: we are at the very beginning of the “Chip on Demand” age. Since the chip is defined by software “bits” we will create chip designers out of software designers this decade, and the number of unique chip implementations (or programs) will skyrocket. Chips will be customized to exactly the particular algorithms or function needed in real time as the data is presented to the chip...without spinning a new chip.

# Example “Chip-on-Demand” Application: Software Defined 3G Wireless Basestation

Chip functionality depends on data traffic



CFox Chameleon Systems 15 0101

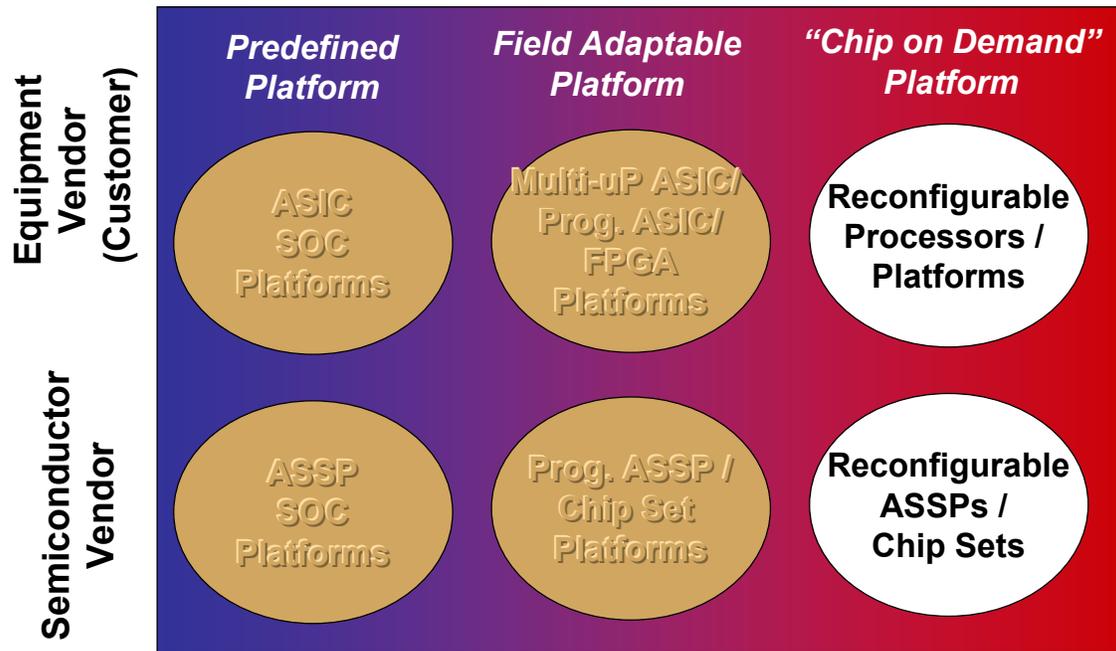
DESIGNCON 2001

An example of this is the next generation cell phone or wireless terminal. Chips for cell phones can now be designed that allow the phone to be “configured” for whatever air-interface is needed through software downloads from the base station as you step off the airplane. As we move to 3G and 4G wireless Web phones the performance requirement and hence the power consumption of the phone increases dramatically. However, with a reconfigurable approach, the functions needed are stored in the base station and not the phone, therefore the overall power consumption is reduced without a corresponding reduction in performance. This is often referred to as the “software-defined radio” approach.

The same approach is being taken in the wireless base station.

# Software Defined Platforms

## Source of Design IP



years months hours secs msec usecs nsecs

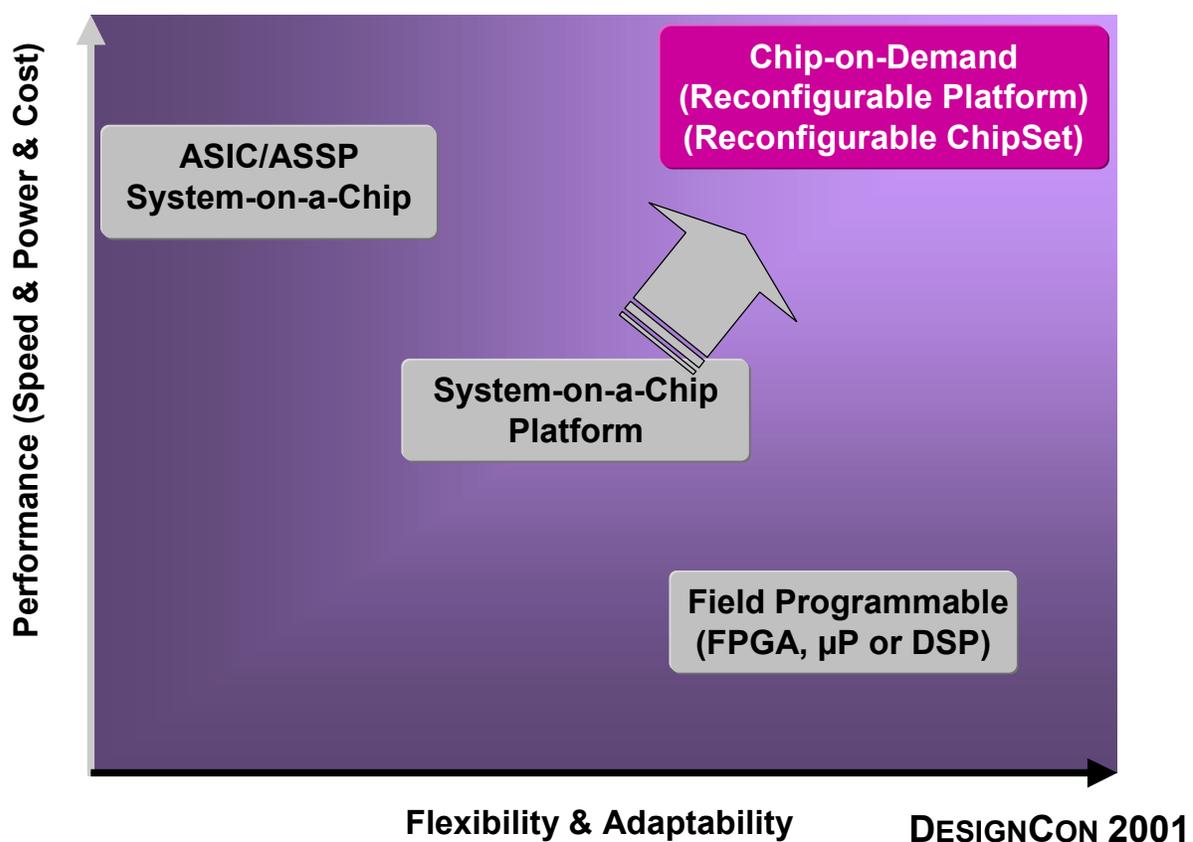
**Speed of Design Change**

**DESIGNCON 2001**

CFox Chameleon Systems 16 0101

The chip-on-demand platform IP can be developed by the semiconductor vendor in which a reconfigurable chip set is produced. This would be the case of the reconfigurable chip for the cell phone example. If the customer design engineer prefers to add value, differentiate, and create their own chip architecture “on-the-fly” then the customer uses design tools provided by the Reconfigurable Platform company. The customer then creates the bitstreams, as many bitstreams as they want, to optimize the application and then determines under what conditions those bitstreams are loaded and removed from the chip. He defines the time-based “schedule” of the IP to be loaded into the chip.

# Design Tradeoffs



CFox Chameleon Systems 17 0101

The net result is that designers that exploit “chip-on-demand” using reconfigurable approaches for the right applications will absolutely break through the traditional performance/time-to-market tradeoffs and achieve increased system performance, reduced system power, fast time-to-market and design flexibility. In some cases it will be the only solution to the problem.

# A Cornerstone of the Coming Era in Field Programmability

## The “Chip on Demand”

CFox Chameleon Systems 18 0101

**DESIGNCON 2001**

This “Chip on Demand” capability of “just-in-time” chip functionality will be a cornerstone of the Coming Era of Field Programmability. Through real-time complete chip software downloads, software designers at equipment makers will effectively be designing deep-submicron chips and they won’t even realize it!