



**HANDSHAKE  
SOLUTIONS**

## **Asynchronous Silicon Compilation**

Symposium 20 years OOTI  
Ad Peeters, March 26, 2009

### **Concurrent Computations and VLSI Circuits Martin Rem, 1984**



- 'VLSI is a medium in which computations can be realized that exhibit a high degree of concurrency.'
- 'Concurrent computations require a very careful design technique, for, as we know, uncontrolled concurrency results in uncontrollable complexity.'
- 'When designing a VLSI circuit we would like to use a suitable program notation for the expression of the computation intended.'
  - It should be 'non-sequential', tailored to concurrent execution
  - Programs should form hierarchical structures

2

### Concurrent Computations and VLSI Circuits Martin Rem, 1984



- Sequential execution and iteration

com  $\text{sem}_1(x, y) : (x; y)^* \text{moc}$

- Parallelism/concurrency

com  $\text{sem}_2(x, y) : x; (x, y)^* \text{moc}$

- Hierarchy

com  $\text{sem}_4(x, y) :$   
sub  $s_0, s_1 : \text{sem}_1$   
 $s_0.y = s_1.x$   
 $(x; s_0.x)^* , (s_1.y; y)^*$   
moc

- No data representation or explicit communication (input/output)

3


### Translating Programs into Delay-Insensitive Circuits Jo Ebergen, 1987



- 'The idea is to realize a component by means of a delay-insensitive connection of basic components.'
  - Specifications distinguish between input and output
  - More direct relation with VLSI circuit realization
- Set of basic building blocks (C-element, Toggle, Arbiter, XOR, Join, ...)
- Formalism for specification of computations/programs
  - Components guarantee certain behavior (on outputs) as long as environment implements certain restrictions (on inputs)
- Correctness criteria for parallel composition based on avoiding interference
- Data-representation not addressed

4

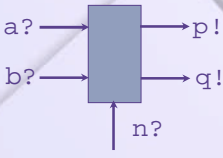
### Translating Programs into Delay-Insensitive Circuits Jo Ebergen, 1987



- Parallel composition of specifications
 

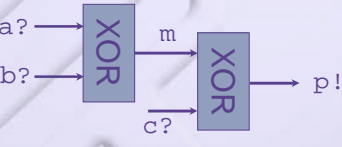
```

pref[a?;p! ]
|| pref[b?;q! ]
|| pref[n?;(p!|q!)]
      
```


- Decomposition of specifications into implementations
 


```

pref[a?;p! | b?;p! | c?;p! ]
→ pref[a?;m! | b?;m! ]
  || pref[m?;p! | c?;p! ]
      
```



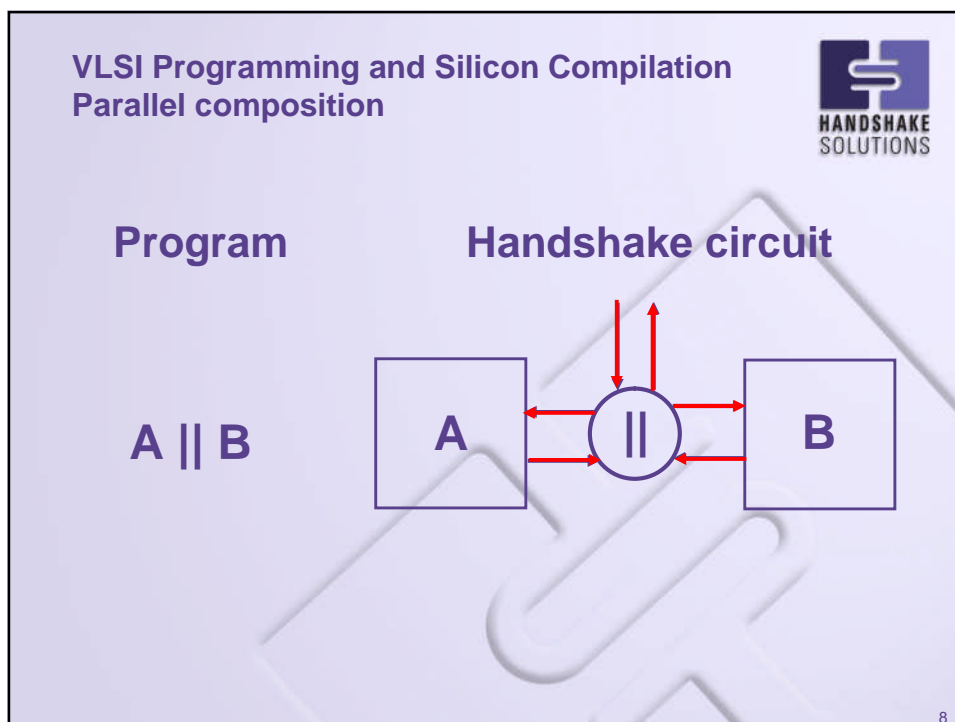
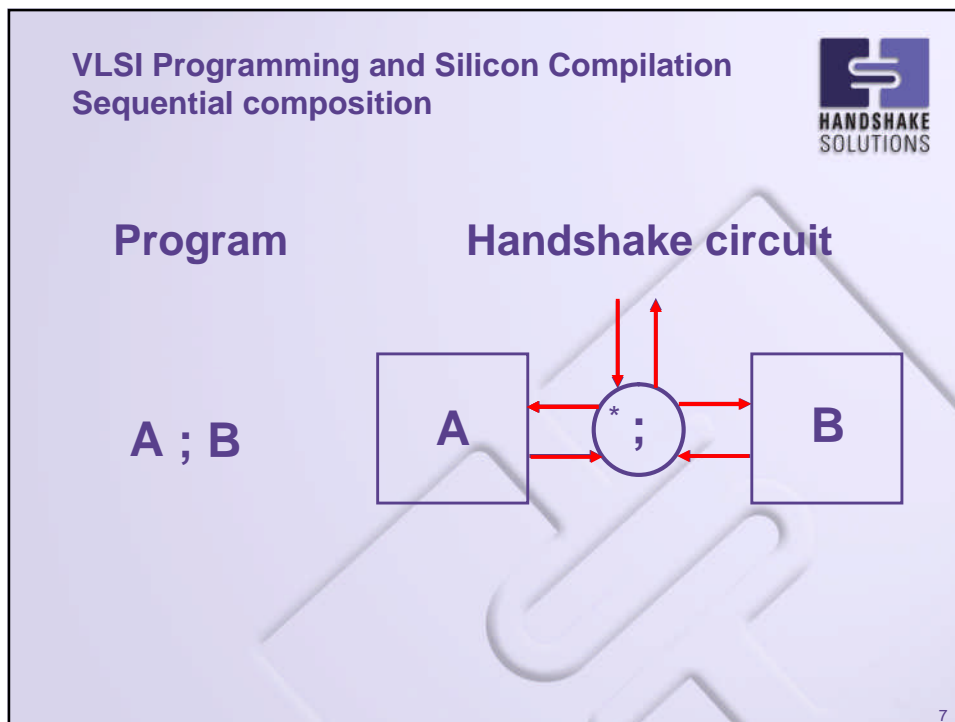
5

### VLSI Programming and Silicon Compilation Kees van Berkel, 1986




- Motivation: to increase the productivity of VLSI design by treating circuit design as a programming activity
- Approach: Compiling programs directly to silicon
- Syntax directed compilation
  - For every syntax construct of the language a component is defined
  - For every component an implementation in VLSI is generated
- Compiler is behavior preserving
  - Parallel composition of the components is equivalent to the meaning of the source program
- 'We chose asynchronous circuits as target for automatic silicon compilation, because asynchronous circuits simplified the translation process and made it easier to take advantage from the abundantly available parallelism in VLSI.'*

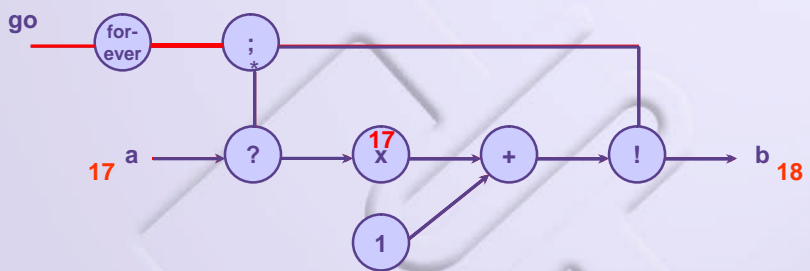
6



VLSI Programming and Silicon Compilation  
Haste and handshake circuits




Top : main proc (a?chan int & b!chan int).  
begin x : var int |  
forever do a?x ; b!x+1 od  
end



9


VLSI Programming and Silicon Compilation  
Handshake protocol implementations



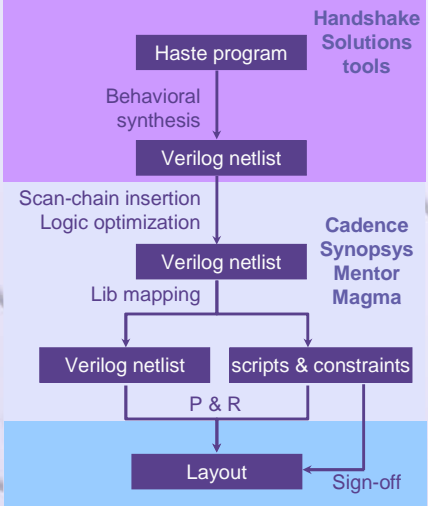
- Number of wires for control
  - $1\omega$  (req and ack on single wire, a.k.a. single-track, tristate)
  - $2\omega$  (separate wire for req and ack)
- Number of phases in protocol
  - $2\phi$  (non return-to-zero, NRZ)
  - $4\phi$  (return-to-zero, RTZ)
  - $\tau$  (synchronous, sampling of req and ack wire)
- Encoding of data
  - double rail (2 wires per bit)
  - single rail (1 wire per bit plus data-valid)
  - M-out-of-N (1-out-of-4 is interesting)

10

### VLSI Programming and Silicon Compilation Timeless Design Environment




- TiDE is a frontend to your existing EDA flow
- TiDE is *complementary to* and *compatible with* third-party EDA tools
- High-level design entry (Haste)
- Standard-cell hand-over
- Scan-test-based Design-for-Test
- FPGA prototyping through synchronous preview of design
- Integrated support for placement and routing, logic optimization and timing sign-off

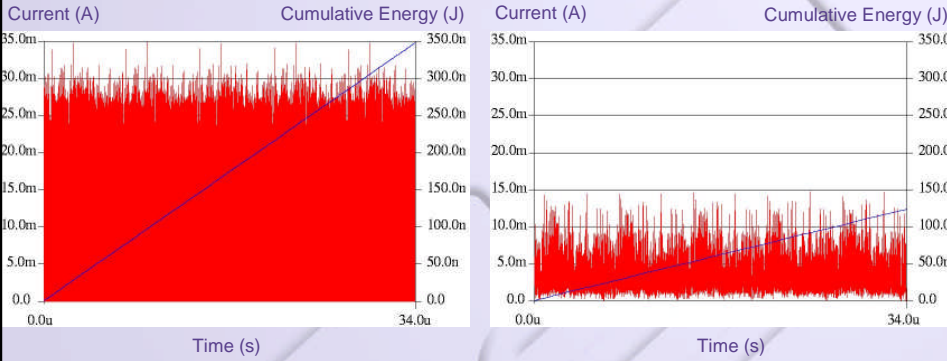


11

### Technology Benefits Low Current Peaks and Total Current

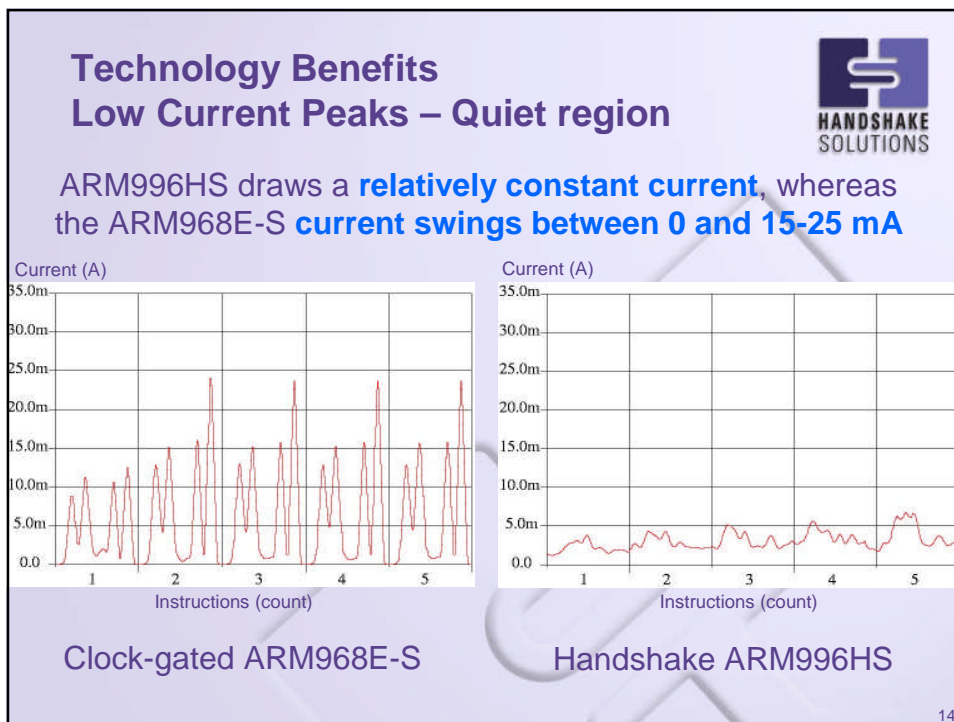
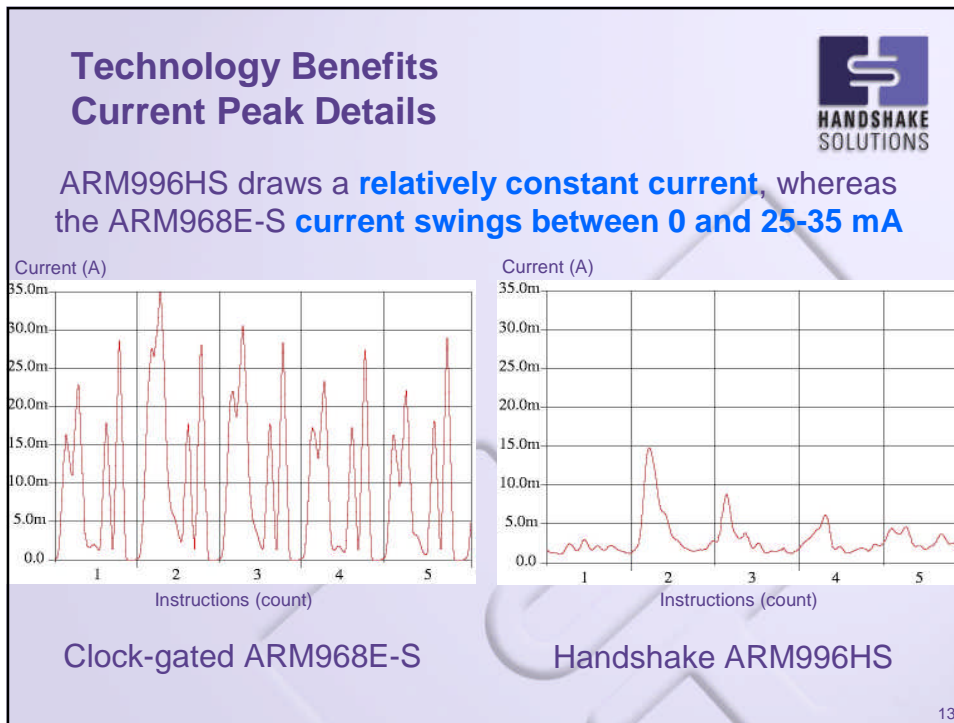


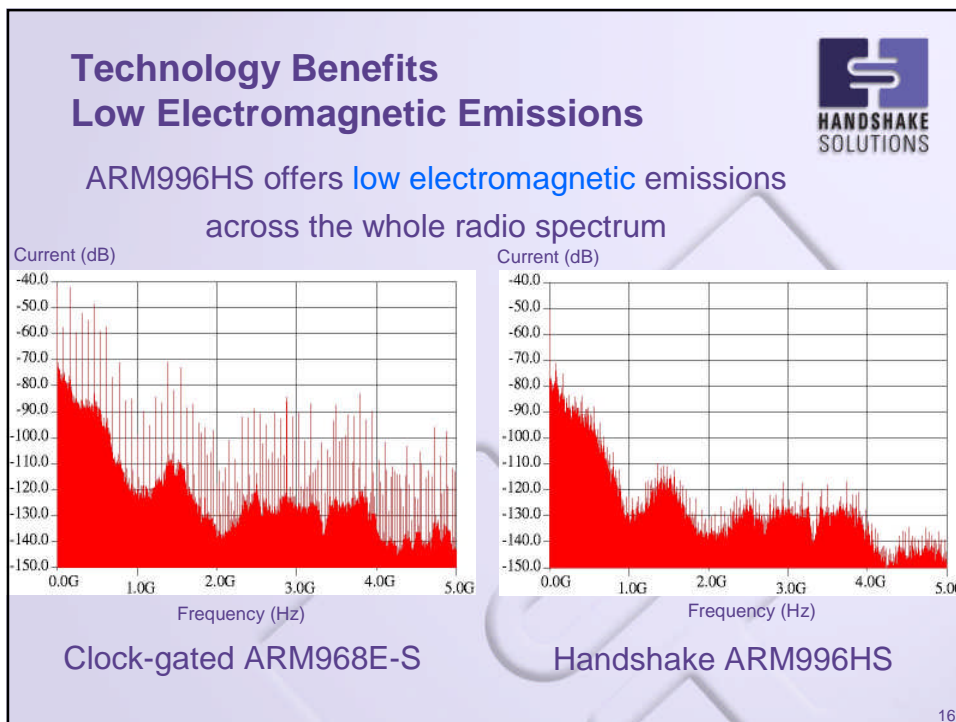
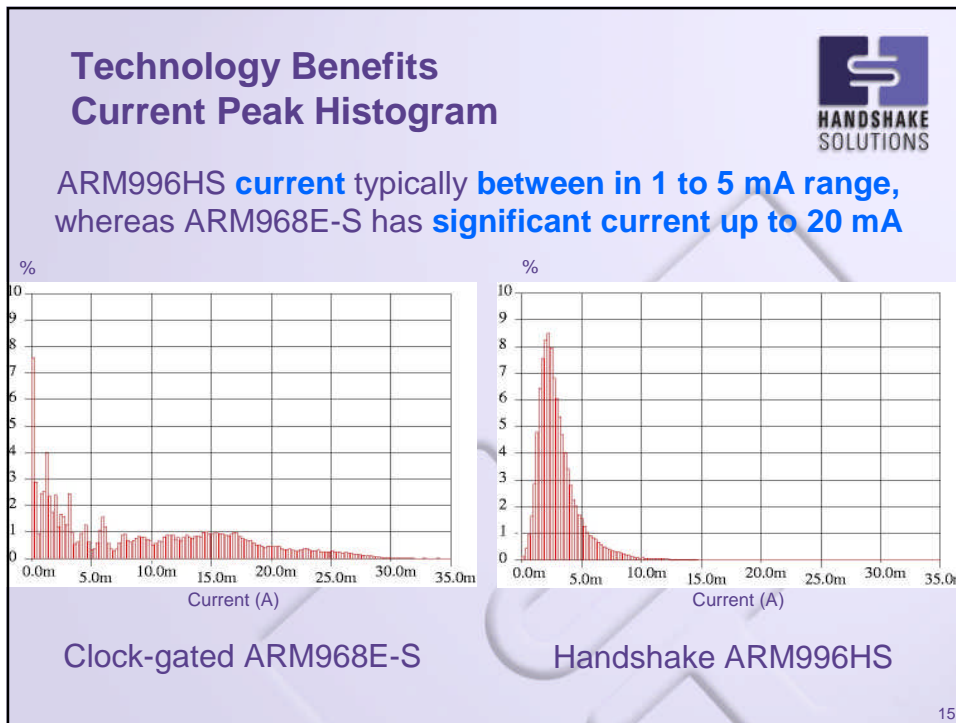
ARM996HS consumes **2.8x less power** than an ARM968E-S and **reduces current peaks by a factor 2.4**



Device	Current Peak (A)	Cumulative Energy (J)
Clock-gated ARM968E-S	~35.0m	~350.0n
Handshake ARM996HS	~14.6m (2.4x lower)	~125.0n (2.8x lower)


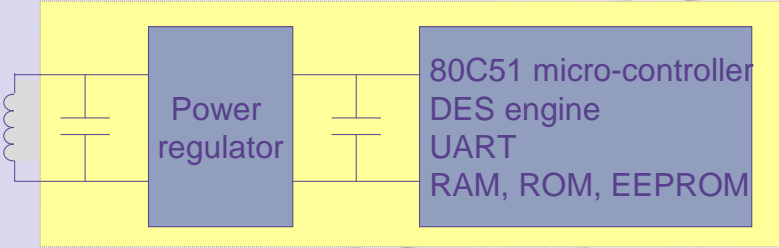
12





## Smart card controllers

Problem: low performance in contactless mode


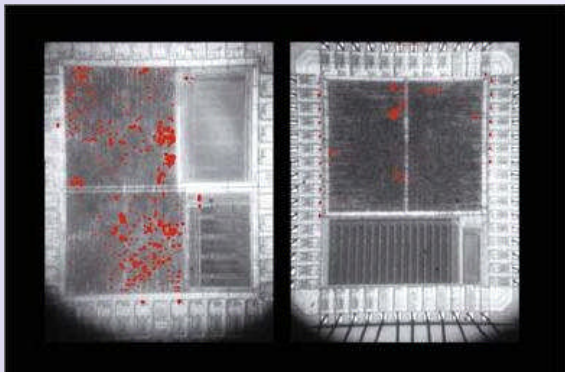



- In contactless mode, smart cards are powered via a coil by a reader
- Reader provides data, timing, and energy (typically at 13.56MHz)
- Energy efficiency of card determines minimum transaction time
- Synchronous 80C51 consumed too much power to complete advanced transactions (including EEPROM update) within 250 msec deadline

17

## Smart card controllers

Analysis: energy consumption 80C51

- Handshake 80C51 (right) is more than 4x more energy efficient than clocked 80C51 (left)
- Additional benefits found:  
lower system cost, more reliable transactions

18

## Smart card controllers

### Solution: low power and more



- NXP Semiconductor's Mifare ProX and SmartMX families
- **Low power**
  - enhanced cryptographic security
  - extra non-volatile memory
- **Low current peaks**
  - low-cost integration with analog
- **Low electromagnetic emission**
  - Better operating distance in contactless mode



19

## Smart card controllers

### Products and derivatives



Energy efficiency enables high performance in contactless operation *and* extra non-volatile memory


- More than 80% of the world's smart passports
- Access control at NASA
- Nokia's 6131 NFC phone




20

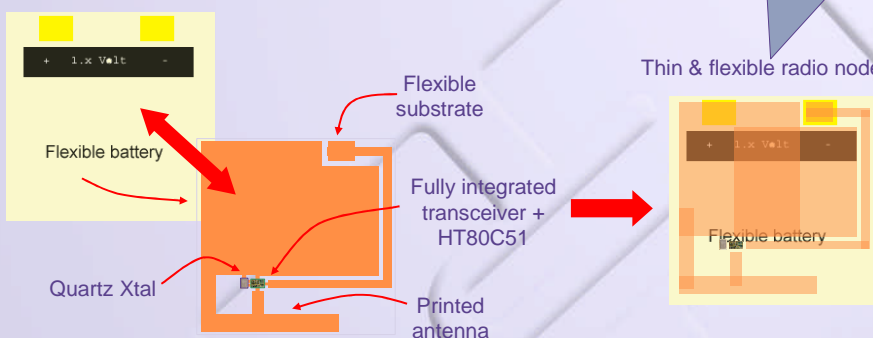
## Flexible active radio

**Challenge: 8051 running on flexible battery**



Low peak current of HT80C51 enables radio operation from flexible battery

Budget was 1mA  
Chip runs at 0.5mA



The diagram illustrates the process of creating a thin and flexible radio node. It starts with a flexible battery (1.5V) and a quartz crystal (Xtal) on a flexible substrate. A fully integrated transceiver and HT80C51 chip are added, along with a printed antenna. The final result is a thin and flexible radio node that maintains the 1.5V battery and the integrated components.

21

## Market success





- TiDE has been used for a range of geometries:
  - 0.8 $\mu$ , 0.6 $\mu$ , 0.35 $\mu$ , 0.25 $\mu$ , 0.18 $\mu$ , 0.14 $\mu$ , 0.13 $\mu$
  - 90nm, 65nm, 45nm
- **More than 750 million ICs with Handshake Technology sold**
- Applications in:
  - Smartcards
  - Automotive
  - Wireless connectivity

22

## Asynchronous Silicon Compilation Lessons learned



- Asynchronous circuits of arbitrary complexity can be reliably realized and brought to market
- Syntax-directed compilation hinders maintainability of optimized code
- Control-driven compilation result in distributed controllers which limits maximum achievable performance
- High-level behavioral design increases the productivity of 'good' designers

23



Asynchronous silicon compilation is fun!

*Thank You*