

What is a Good Platform?

Axel Jantsch

Royal Institute of Technology, Stockholm

February 11, 2003

What is a SoC Platform

- Communication infrastructure
- Resource management services
- Design methodology and tools



Platform Characteristics

- Tradeoff between efficiency and cost
- Application area specific
- Guarantees and Predictability
 - ★ Platform inherent guarantees
 - ★ Static guarantees
 - ★ Dynamic guarantees
 - ★ E.g. communication guarantees
 - * Delivery
 - * Minimum bandwidth and maximum delay
- Scalability

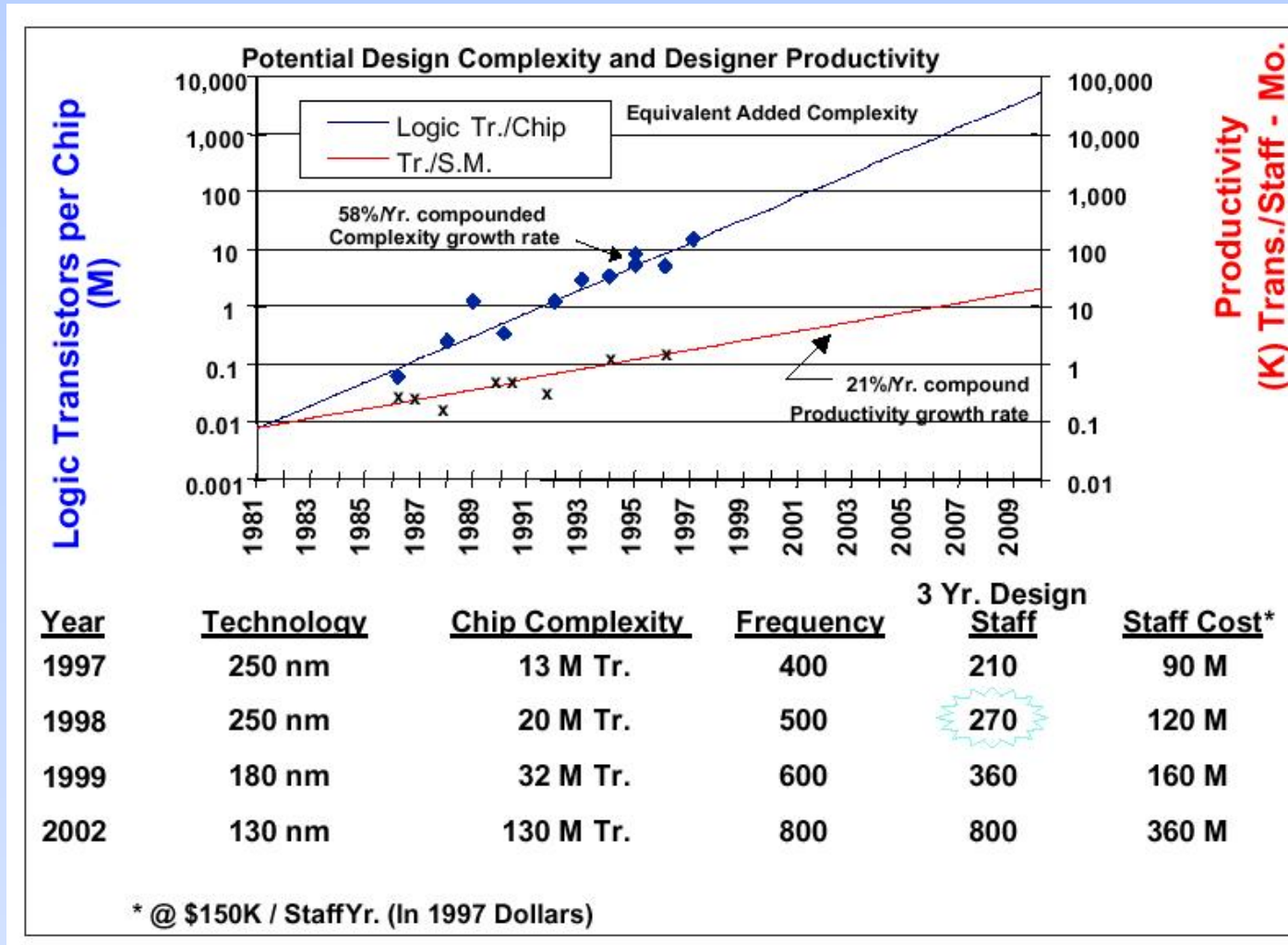


Scalability of a Platform

- Performance - Cost
- Size
- Reliability
- Design methodology



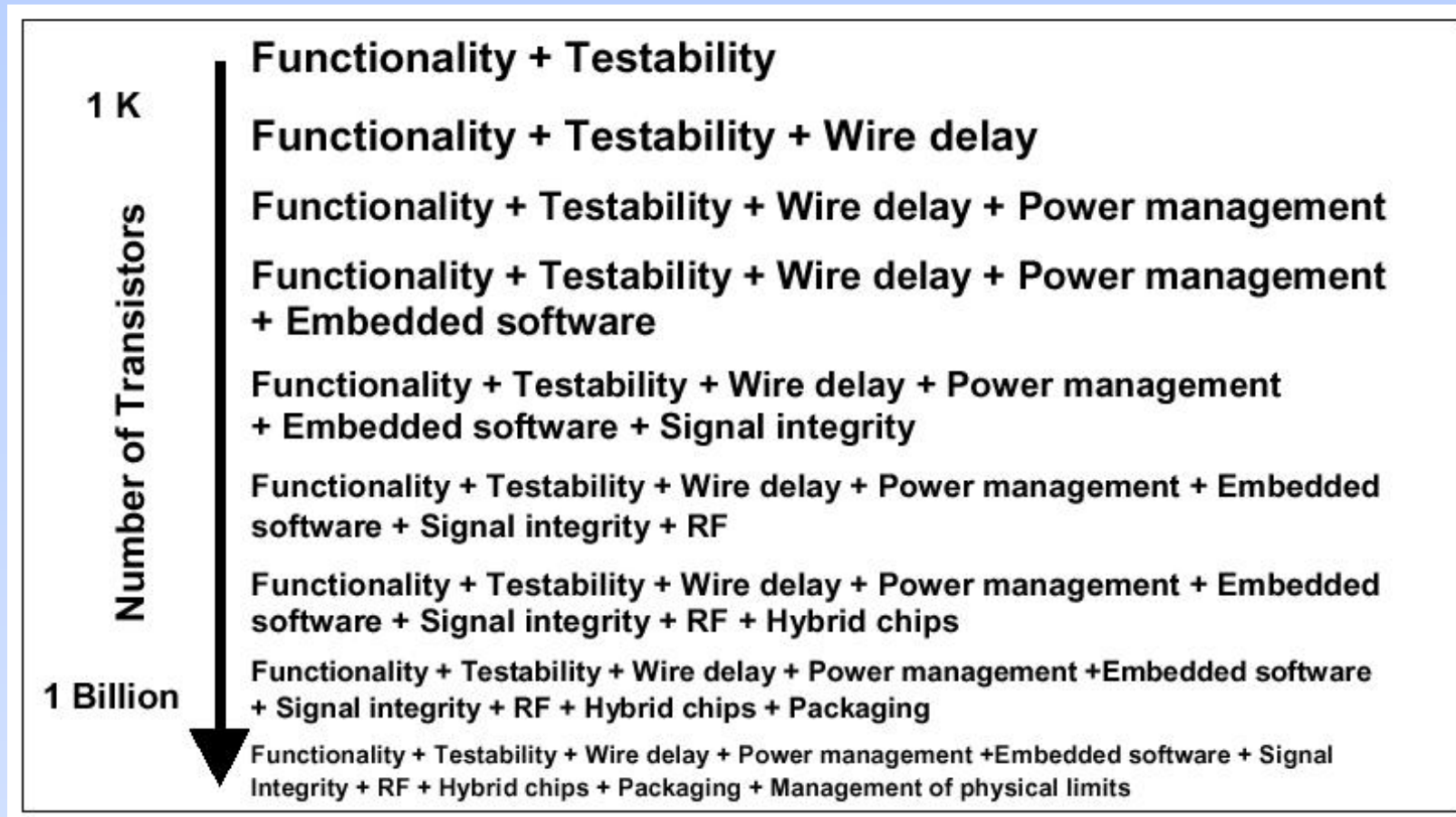
Design Productivity Gap



International Technology Roadmap for Semiconductors 1999



Super-exponential Increasing Design Complexity



International Technology Roadmap for Semiconductors 1999



Arbitrary Composability

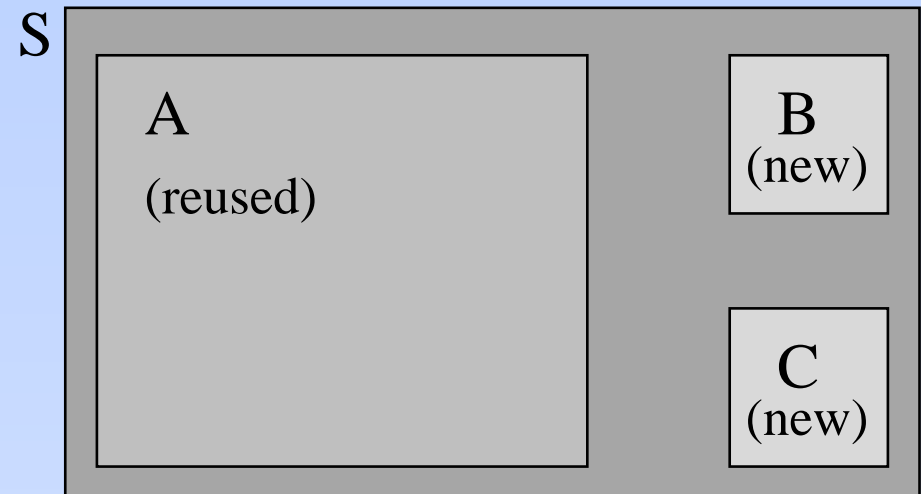
Given a set of components C and combinators O .

Let A_1 be a component assemblage.

(C, O) is **arbitrary composable** if

$$A_1 + B \Rightarrow A_2$$

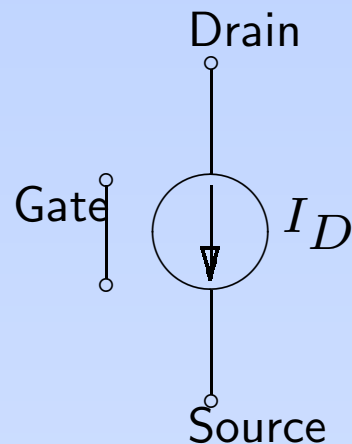
can be done for any $B \in C$, $+ \in O$ **without changing the relevant behaviour of A_1** .



A MOS Transistor Model

$V_{DS} > V_{GS} - V_T$ (conducting state):

$$I_D = \frac{k'_n W W}{2L} (V_{GS} - V_T)^2 (1 + \lambda V_{DS})$$



$V_{DS} < V_{GS} - V_T$ (sub-threshold state):

$$I_D = \frac{k'_n W W}{L} \left((V_{GS} - V_T) V_{DS} - \frac{V_{DS}^2}{2} \right)$$

where

$$V_T = V_{T0} + \gamma (\sqrt{|-2\phi_F + V_{SB}|} - \sqrt{|-2\phi_F|})$$

V_{DS} ... drain-source voltage

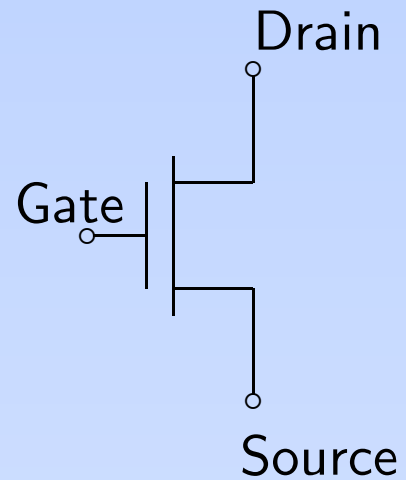
V_{GS} ... gate-source voltage

V_T ... threshold voltage

I_D ... drain-source current

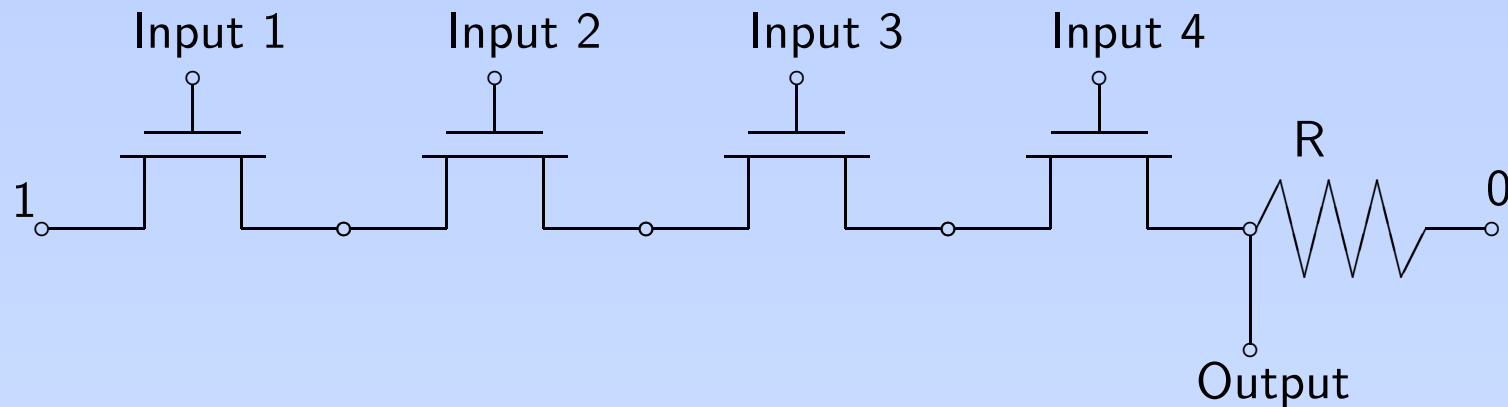


A Transistor as Switch



| Gate | Drain | Source |
|------|-------|-----------|
| 0 | 0 | undefined |
| 0 | 1 | undefined |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

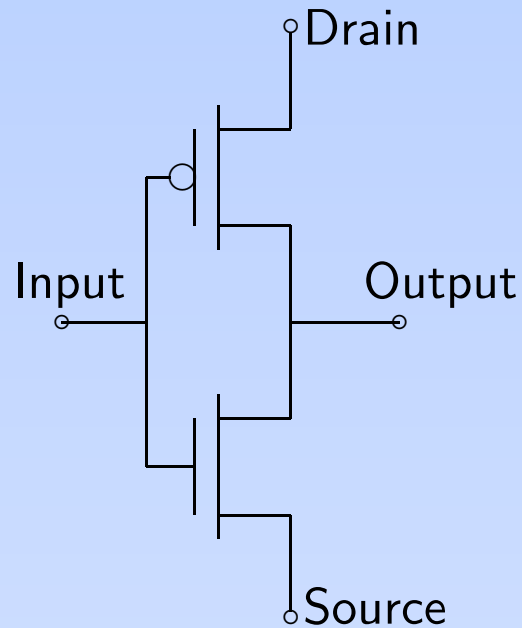
An AND Gate as Transistor Network



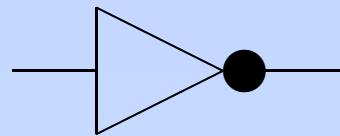
Two problems with arbitrary transistor networks:

- Output is not defined when input is 0.
- Voltage drop between drain and source is relevant but not visible.

An Inverter as Transistor Network



(a)



(b)

| Input | Output |
|-------|--------|
| 1 | 0 |
| 0 | 1 |

(c)

Gate Based Abstraction Level

1. The primitive elements are defined by simple models, i.e. small truth tables in this case.
2. The primitive elements can be implemented in a wide range of technologies.
3. The model holds even for arbitrarily large networks of primitive elements.



Platform and Composability

- A good platform has the arbitrary composability property.
- There are building blocks that can be added without changing the rest of the system.
- The building blocks can be:
 - ★ Computation resources
 - ★ Communication resources
 - ★ Storage resources
 - ★ I/O resources
 - ★ Resource manager modules (Scheduler, OS, ...)
 - ★ Features: Resources + System functionality
- The “relevant behaviour” includes functionality, performance, cost, reliability, power consumption.
- \implies We can make guarantees.



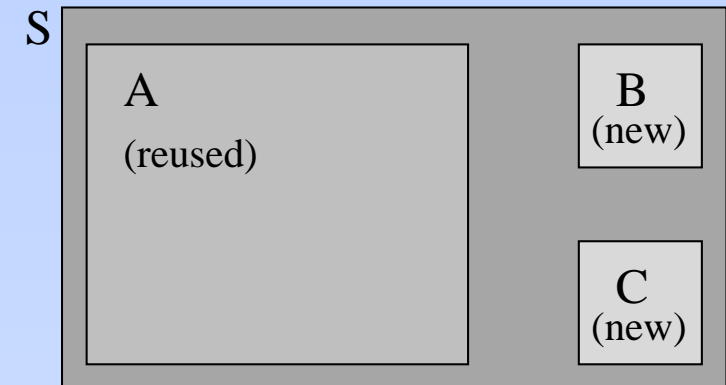
Linear Effort Property

Given a set of components C and combinators O .

Let A_1, \dots, A_n be component assemblages.

A **design process** using C and O to build a system has the **linear effort property** if A_1, \dots, A_n can be integrated into a system S with an effort dependent on n but not on the size of the assemblages: $I_{\text{effort}}(n)$.

Total design effort for S is



$$\text{Deffort}(S) = \text{Deffort}(A_1) + \dots + \text{Deffort}(A_n) + I_{\text{effort}}(n)$$

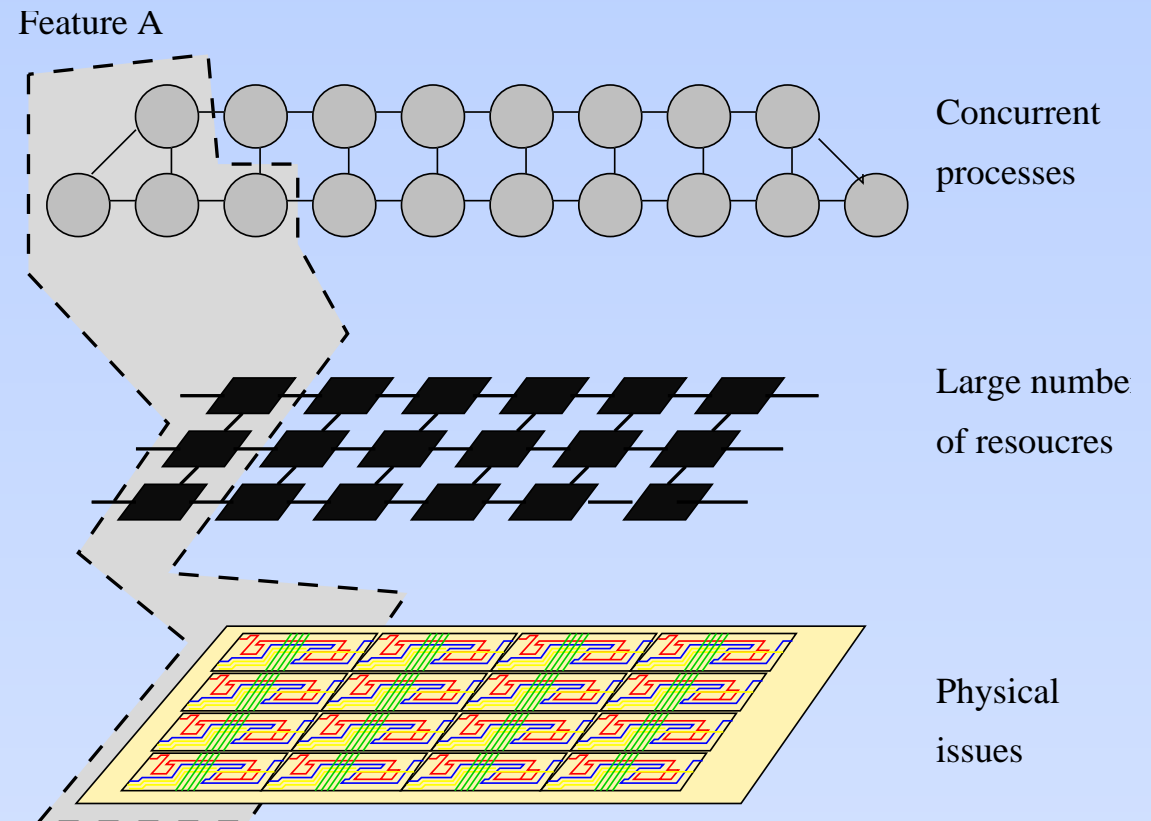
Methodology and Linear Effort

- A good platform comes with a methodology that has the linear effort property.
- The platform is then scalable with respect to capacity increase by reusing every bigger components.
- This implies an invariance with respect to hierarchy: Composition works as well for primitive components as for arbitrary assemblages.



A Network-on-Chip Based Design Process

- Configuring the platform
- Selecting resources
- Reuse of features
- Evaluation and integration



Summary

- A good Platform greatly restricts the design space.
- It trades in optimality for design efficiency and predictability.
- The arbitrary composability and the linear effort properties provide a scalable platform.
- The reuse of ever bigger assemblages and components is platform inherent.
- Predictability of functionality, performance, cost, power consumption and reliability is a prerequisite as well as a consequence for the arbitrary composability and the linear effort properties.

