

Progresivní technologie v informatice II

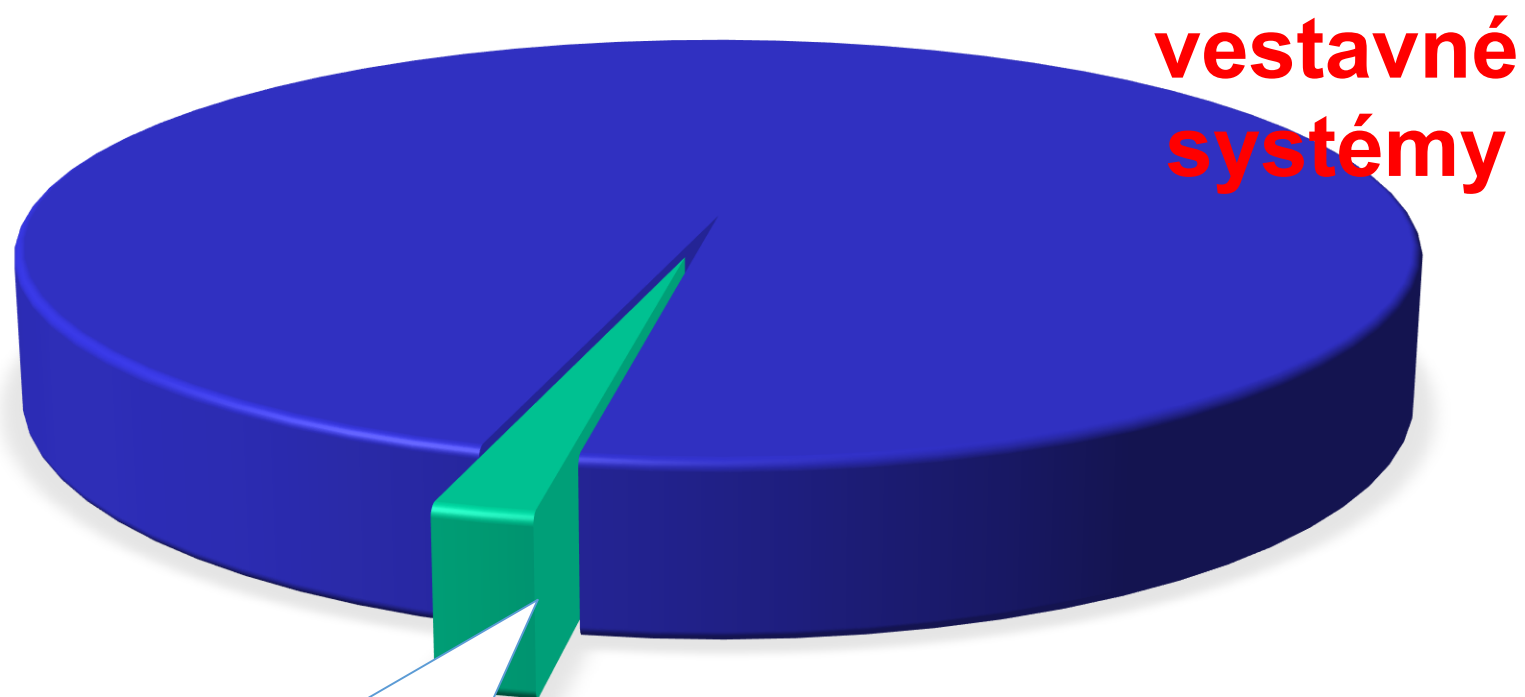
Vestavné systémy

návrh “chytrých” zařízení, požadavky, vlastnosti a využití

Fakulta informačních technologií
České vysoké učení technické v Praze

Co znamená „*embedded*“

- Vestavné systémy („embedded“)
- Systémy na čipu (SoC)
 - Integruje všechny komponenty počítače (elektronického systému) do jednoho IO “integrovaného” obvodu
 - Většinou obsahuje číslicové, analogové, mixed-signal bloky, DSP součástky, periferie, paměti
- Omezující požadavky
- Redukce ceny, optimalizace
- Jsou všude: „smart devices“



**počítače, tablety
2%**



**všechna tato zařízení jsou ovládána zabudovanými
(= vestavěnými počítači)**



Struktura

- **Procesor (nebo více procesorů) ... ASIP** (application specific integrated circuit/zákaznický integrovaný obvod), **DSP (digital signal processor)**
- **Paměťové bloky ... RAM, ROM, EEPROM, flash, ...**
- **Zdroje časování (oscilátory, fázové smyčky)**
- **Čítače-časovače, real-timové časovače, power-on reset generátory**
- **Analogový interface ... AD-DA převodníky**
- **Regulátory napětí a příkonu**
- **Propojení**
- **Externí interface ... USB, FireWire, Ethernet, USART, SPL**
- **Standardizace**

Struktura a návrh

- Je nutné navrhnout hardware i software + spojení s okolím
- SW ... řídí procesorová jádra, periferie a interface



- Paralelně HW i SW (hardware-software codesign)
- Často skládání HW bloků, jader, driverů ... problém CAD nástrojů
- Blok programovatelného/rekonfigurovatelného hardwaru – FPGA
- Verifikace návrhu – funkční (HDL, SystemVerilog, SystemC, OpenVera), formální (UVM)

Omezení

Podle požadavků aplikace:

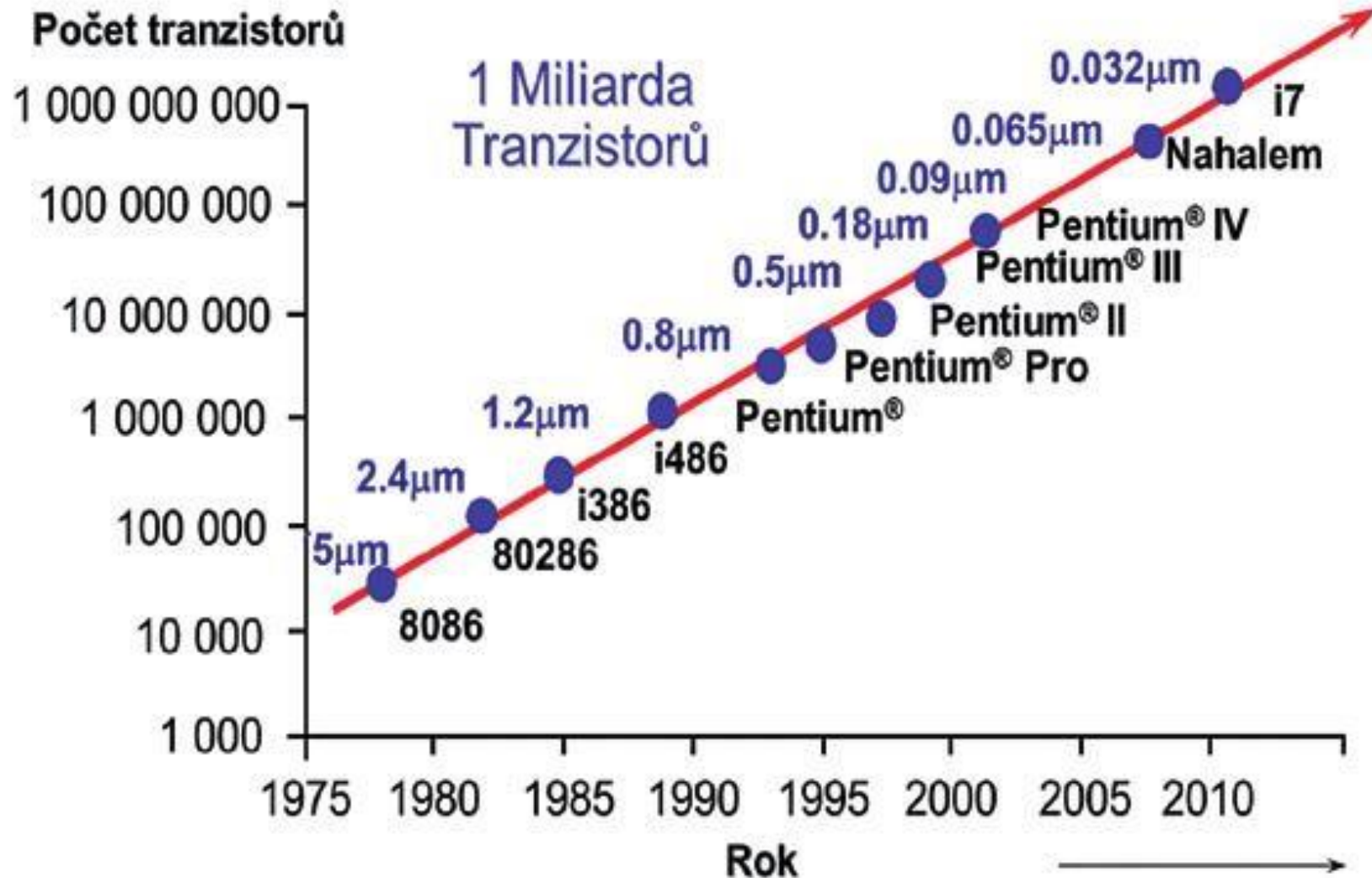
- Technologie:
 - ASIC – plně zákaznický návrh
 - Standardní bloky – jádra, mikrokontroléry, program v C
 - Programovatelný HW – FPGA
- Omezení
 - Velikost (area-overhead)
 - Spotřeba ... low-power design
 - Real-time
 - Vyšší spolehlivost ... lepší konkrétní parametr podle aplikace
- Nižší cena a „time-to-market“

- Rychlý rozvoj technologií ... Mooreův zákon
 - Možnost realizovat a prakticky používat složitější systémy
 - Možnost implementovat adaptivní systémy (= složitější algoritmy, strojové učení)
 - Nové technologie, budoucnost, vize
 - Open source
-
- Vhodné modely a návrhové prostředky



Moorův zákon ... > 40 let!

Evoluce mikroprocesorů



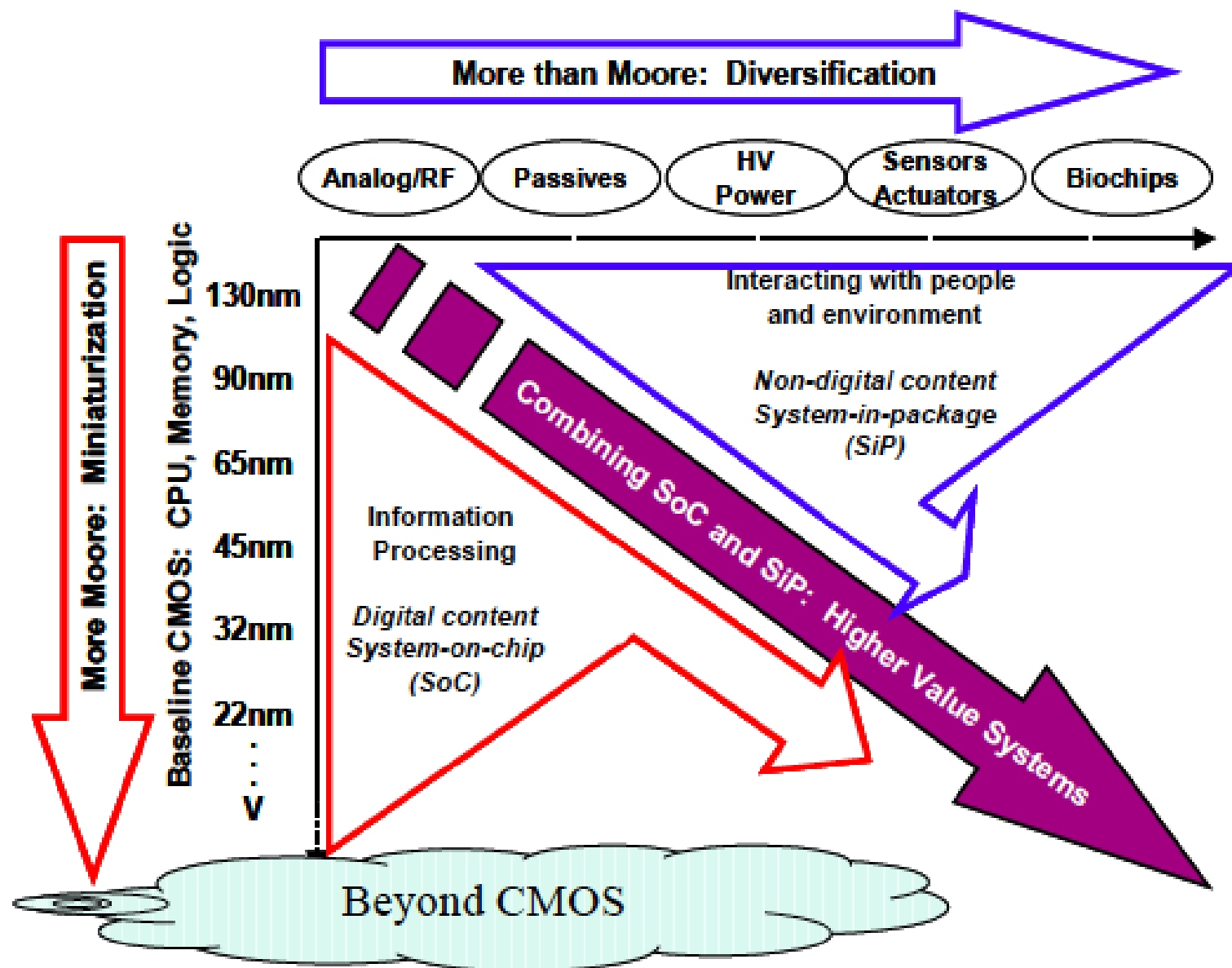


Figure 5 Moore's Law and More



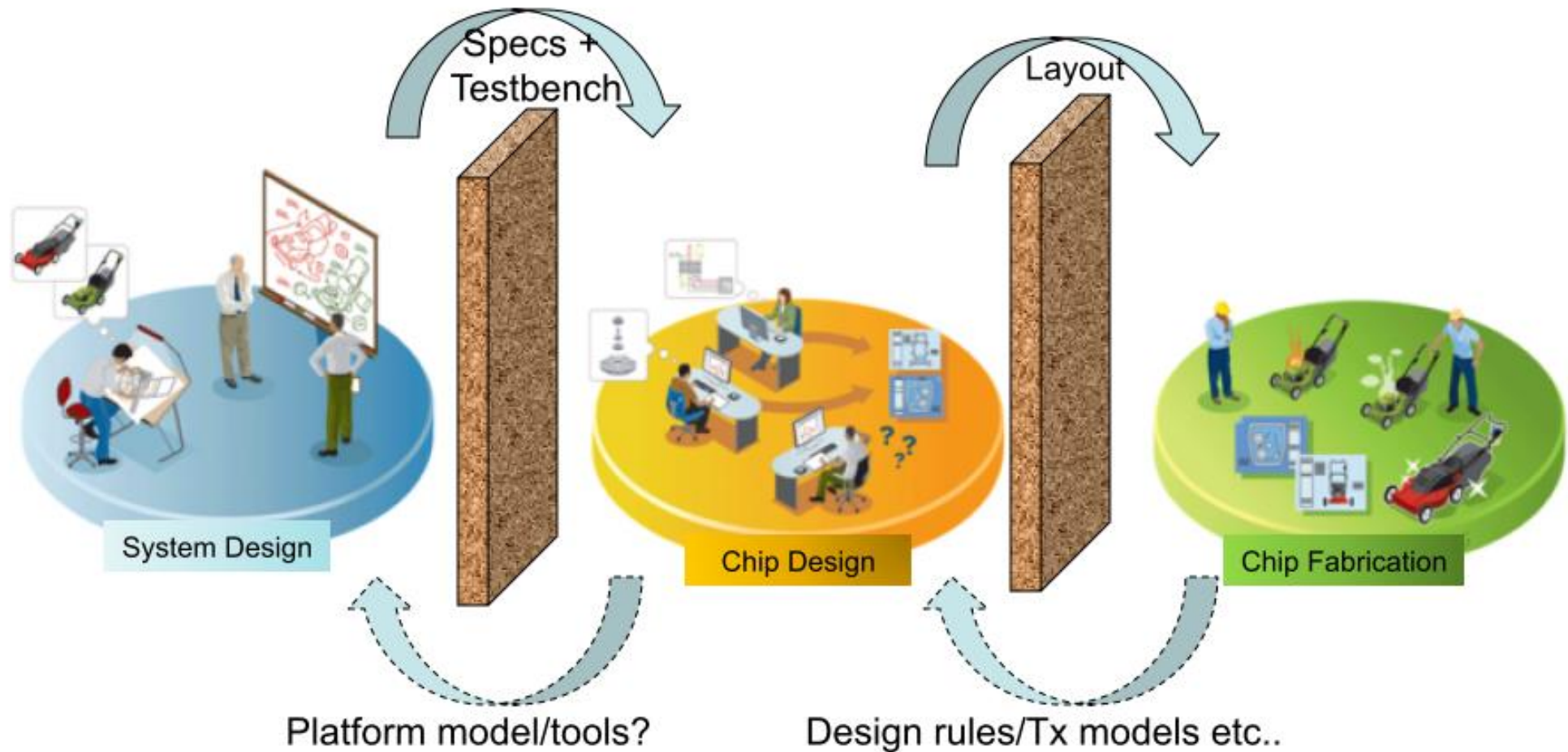
kniha

CHIPS 2020

A Guide to the Future of Nanoelectronics

<https://www.chips2020.net/list-of-chapters>

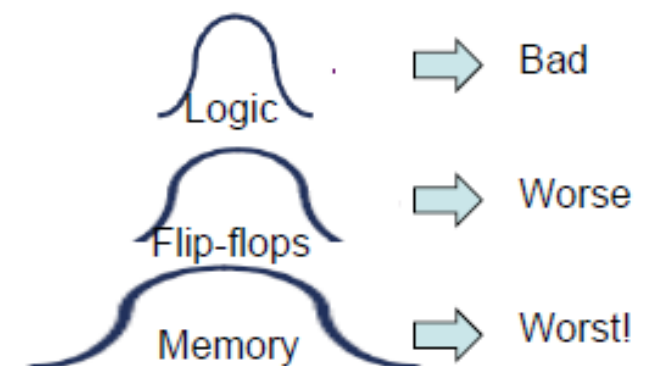
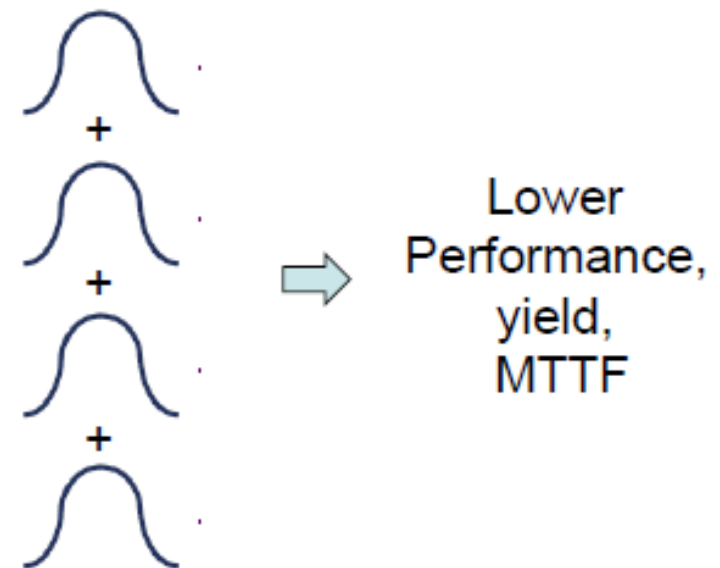
Challenge: Today's design process– The Walls



- Specs in Matlab or C
- Chip output must match testbench **bit-exactly**
- However, those testbenches reflect a small subset of use cases

- Transistors on a chip are no longer identical
- Smaller features, low supply voltage make circuits more susceptible to noise
- Models used are increasingly inaccurate
- Temperature, etc...

Becoming more acute with advanced fabrication technologies

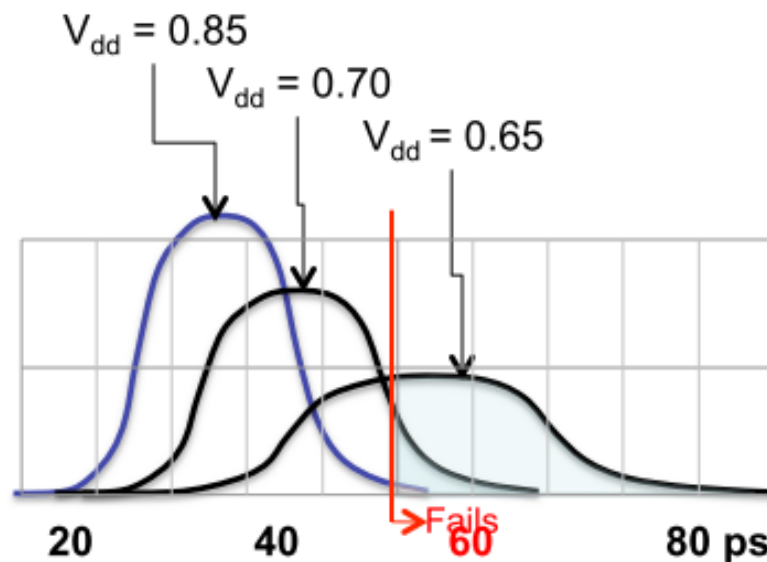




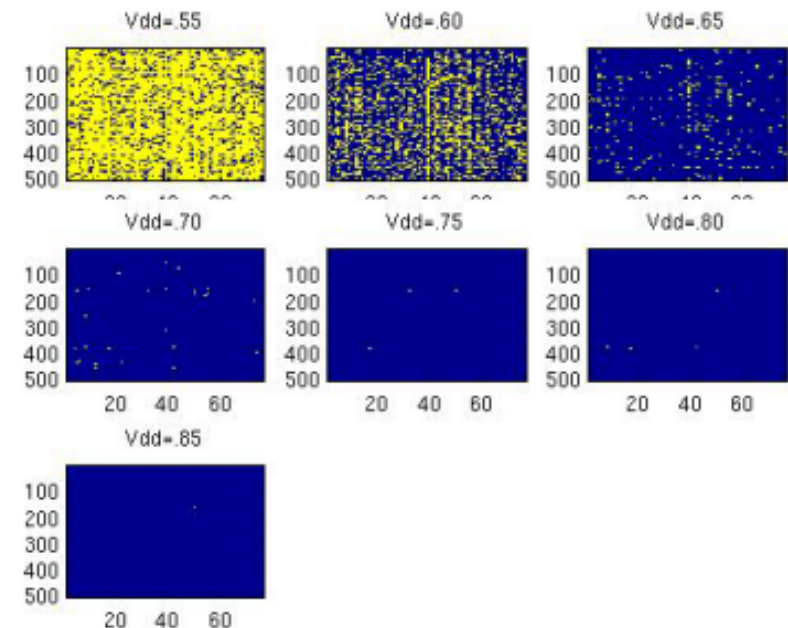
Conventional wisdom: when you reduce V_{dd} , you should run slower.
What happens if I reduce V_{dd} and keep running at the same frequency?

Take a memory array, for example:

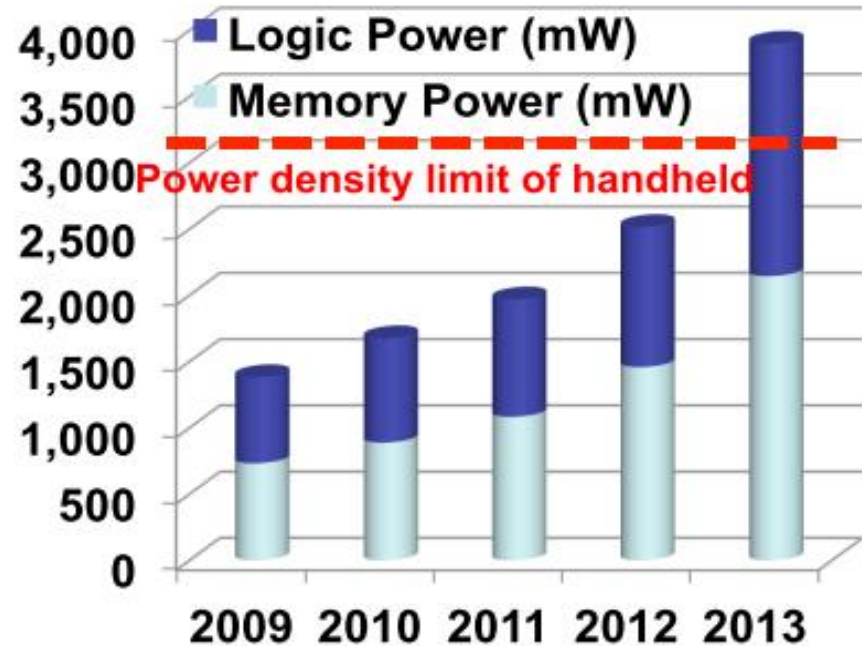
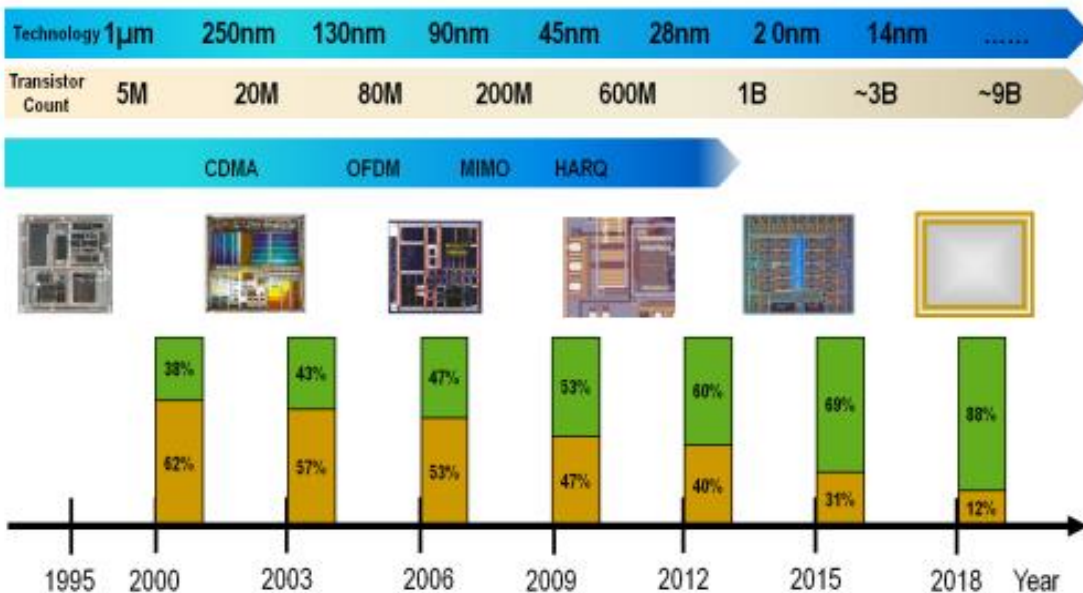
- Variability -> variable read, write delay across memory cells
- V_{dd} reduced -> increased probability of access failure/cell -> more cells failing



Memory Access time

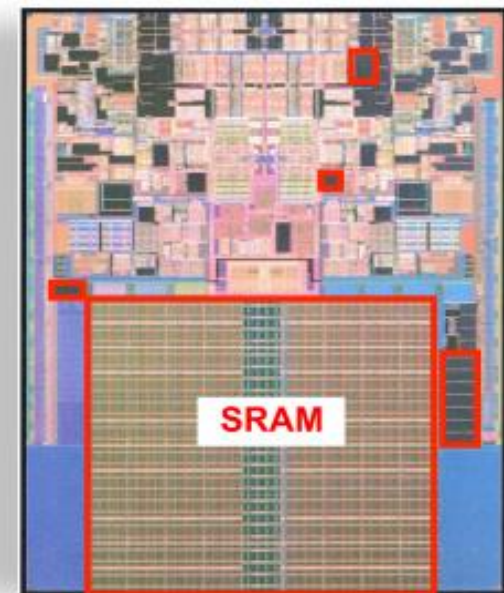


Observation: SoC Becoming Memory Dominated



Focus on memories
Logic later

- On chip SRAM contains 50-90% of total transistor count
 - Xeon: 48M/110M
 - Itanium 2: 144M/220M
- SRAM is a major source of chip power dissipation



(Picture courtesy of Intel)
Intel Penryn™



Fixed (trvalé)

- Manufacturing errors
- Predominant in above 100nm technologies
- Redundancy solutions

Transient (přechodné – dočasné)

- Alfa částice (SEU)

Operating condition (provozní podmínky)

- Voltage, frequency, temperature
- Predominant in sub 100nm technologies
- Defects are due to :
 - Gate Length Variation (GLV)
 - Random Dopant Fluctuation (RDF)

- Manifest themselves at the circuit level as inter-die variation in V_{th} (threshold voltage - hraniční napětí)



- Jak se bude chovat vestavná paměť při agresivním snížení napětí?
- Jak kompenzovat chyby paměti na systémové úrovni?
- Kolik ušetřím energie? Co mě to bude stát?
- Snížením V_{dd} se zvyšuje pravděpodobnost poruchy ...
- Tolerance chyb za cenu redundance
- S nespolehlivostí je třeba počítat již na **systémové, tzn. softwarové úrovni**



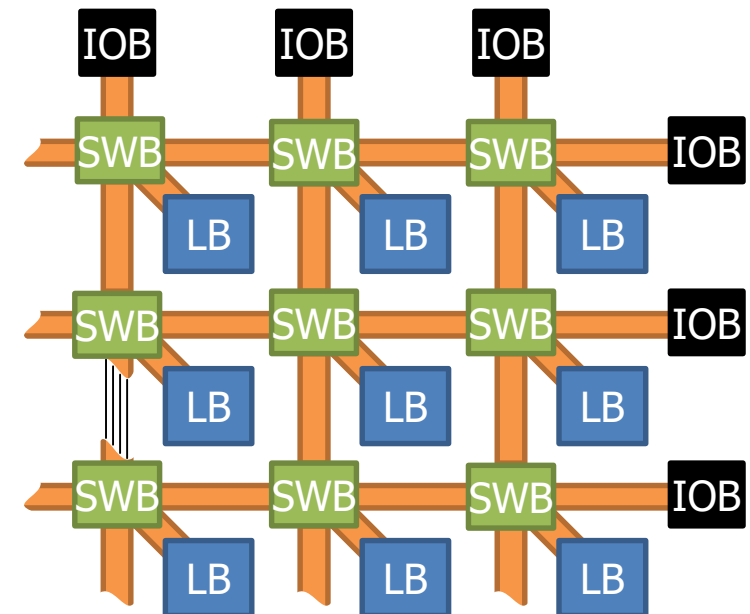
- *Počítej s (ne)spolehlivostí*: systémový návrhář by měl předpokládat určitou úroveň spolehlivosti
- a
- *Vyjádři (ne)spolehlivost*: návrhář SoC by měl umět modelovat spolehlivost na systémové/softwareové úrovni

Otázky:

- Co je hardware a co je software?
- A kam patří FPGA?
- A je všechno digitální?
- Co je cyber-physical systém (CPS)?



- FPGA ... pole programovatelných „hradel“
- FPGA „RAM/flash based“
- Rekonfigurovatelná implementace HW:
 - Konfigurační data – bitstream
 - Definuje implementovanou funkci
 - Definuje strukturu obvodu
- **Garance** úrovně spolehlivosti
- Poruchy a jejich modely
„stuck-at faults“ nepokrývají možné defekty v FPGA



→ jiné modely než ASIC



- Pracuje paralelně, stále *vyšší výkon, ale i komplikace návrhu*
- Cílové prostředí není lineární sekvence instrukcí, ale „2,5D“ prostor (více 2D ploch navrstvených) *vyšší výkon, ale i komplikace návrhu*
- Prostorové vztahy hrají roli
- Předmět počítačové podpory
 - EDA: Electronic Design Automation
 - architektury SW
 - algoritmy



- Je sekvenční
- Pracuje podle instrukčního cyklu
- Rekonfigurace = jiný program
- Paralelismus ???

Co chceme a kdy a jak nejlépe



vykolíkování hřiště vs holistický přístup



Open Source Hardware

Open Source Software - Definition

"OSS is licensed software in which the source code is made available to users to enable them to modify it for their own purposes and (within certain restrictions) redistribute original and derived works as they see fit."

- No one has exclusive control over the term "open source"
- Not an enforceable copyrighted term or trademark
- Open Source Initiative (OSI) www.open-source.org – was founded in 1998 & has unofficial power over the core concepts

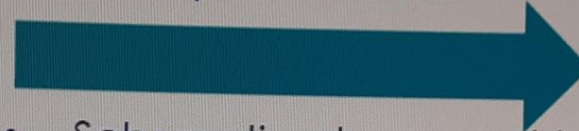


Source: Derivis, "Learn the Basic Principles of Open Source Software", 10 Nov 2008 ID# 00014771

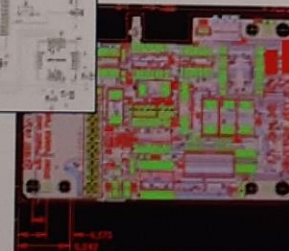


From Open Source Software ...

Hardware design files are
Open Sourced



- Schematics, topographies
- BOM, Drill, Pick and Place, ...
- HDL sources
- Mechanical drawing
- Firmware (sources & bitstreams)



...to Open Source Hardware

THALES

z keynotu [Euromicro DSD/SEAA 2020](#)

Prof. Dr. Cédric Jean-Jacques Demeure

Some trends in CPS systems and underlying technologies



- Propojení na čipu je limitujícím faktorem pro zvýšení výkonu a snížení spotřeby
- Mooreův zákon – stále se zvyšující hustota integrace
- Vede i k vyšší pravděpodobnosti chyb
- Nemožnost použít jeden rozvod CLK – problémy synchronizace ... NI-EHW
 - Globálně asynchronní / lokálně synchronní (GALS)
 - Několik různých hodinových domén
- Možné řešení ... využití síťových principů i na čipu = NoC ale s ohledem na omezení (spolehlivost, velikost, spotřebu, výkon podle aplikace)



Fyzická úroveň

- Komunikační kanály jsou realizovány na čipu pomocí propojovacích vodičů
- Bezchybnou komunikaci ale není možné garantovat (šum, SEU, ...)

Úroveň architektury a řízení

- Architektura určí topologii NoC a fyzickou organizaci
- Protokoly specifikují jak využít síťové zdroje ve funkčním režimu systému
- Rozdíl mezi mikro-sítí (NoC) a obecnou sítí: **nutnost splnit omezující energetické požadavky**

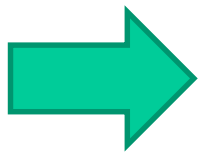
Využití různých architektur propojení (sdílené prostředky, přímé, nepřímé i hybridní sítě)



2 hlavní cíle:

- Zajištění portability a obecnosti aplikací mezi různými platformami ... standardizace
- Poskytnutí nějaké inteligence pro efektivní využití distribuovanosti platformy ... modely

- Jaký použít nástroj pro dorozumění mezi návrhářem a zadavatelem?
- Jak splnit požadavky pro různá omezení (velikost, spotřebu, spolehlivostní parametry a které to jsou, cenu, ...)?
- Jak optimálně realizovat systém obsahující HW, SW, a další bloky?
- Jak tyto požadavky garantovat současně ve vývoji, výrobě a při používání systému?



Je třeba vhodný model

.... ale každý model je zjednodušený



Hierarchický popis

- Celek je složen z částí, části z menších částí... → **úrovně hierarchie**
- Složitost popisu na každé úrovni dána schopností lidí (a strojů) vnímat (žádná funkce nemá mít víc jak 500 řádek, každé schéma se musí vejít na papír A2, ...)
- Odlišení domén chování, struktury a fyzické implementace
- Dekompozice obvodu (*paralelně*)
- Dekompozice algoritmů (*sériově*)
- Analytické a syntetické kroky
- Důkladnější algoritmy (*jinak to prostě nejde*)

• CAD systémy, EDA tools

- Von Neumann model (Sekvenční provádění, paměť programu, etc.)
- Discrete-event model, časový diagram ... VHDL
- Konečný automat (Finite state machine FSM) ... klasický (jednoduchý) nebo „state charts“ [Harel, 1987] s hierarchií
- Diferenciální rovnice



Takový jazyk, který toto všechno splňuje neexistuje



je třeba kompromis

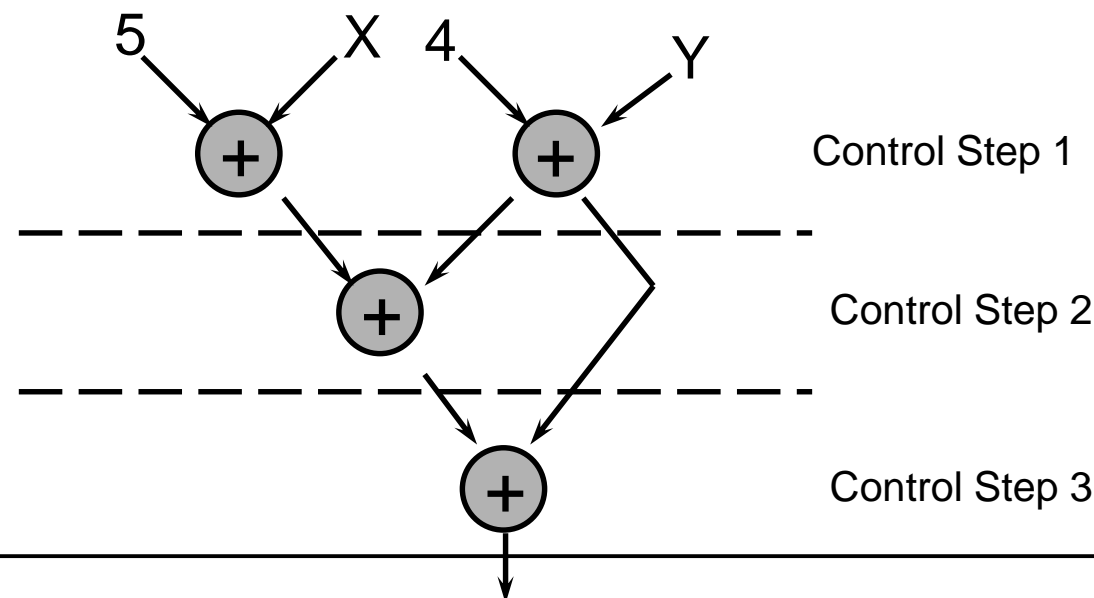


- Specification and Description Language (SDL) i pro distribuované systémy
- Dataflow models:
 - zcela odlišný způsob náhledu na výpočet : imperativní jazykové styly, pohyb dat je prioritou, systém (ne programátor) zodpovídá za plánování
- Kahn Process Network
- Synchronous Dataflow (SDF)
- Objektově-orientované reprezentace
- Petri Nets a jejich rozšíření (Timing PN, CPN, GPN, ...)



- Grafy obsahující uzly odpovídající operacím ... buď HW nebo SW
- Obvykle využíváné v high-level HW syntéze
- Modelují data flow, řídicí kroky, souběžné operace

Příklad:





- Používají techniky dříve aplikované jen na SW i pro správu HW
- Využití jazyka C++ pro popis HW a vyjádření OO vlastností
- Využití konceptů OO:
 - Datové abstrakce
 - Skrývání informace
 - Dědičnost
- Využití stavebních bloků pro využití OO podpory
 - Opakované využívání komponent
 - Nižší cena návrhu
 - Rychlejší návrhový proces
- Zvýšená spolehlivost

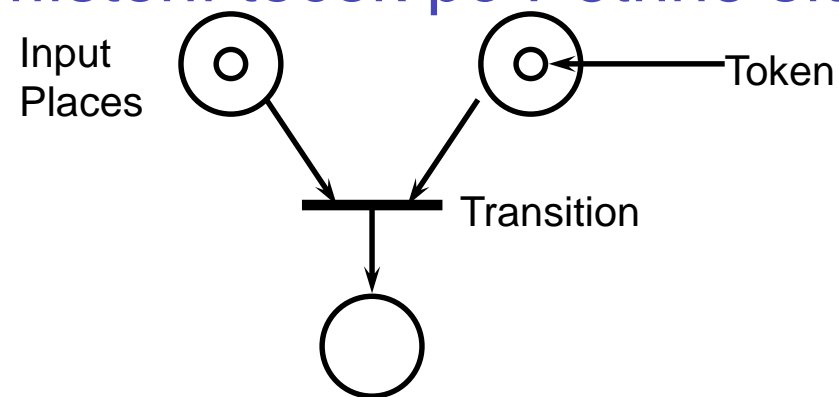


- Systems Modelling Language SysML
- je určen pro modelování SW částí (nejen) vestavných systémů
- vznikl jako varianta UML, ale od verze 2.0 je nezávislý
- využívá 7 ze 14 UML diagramů + přidává dva nové

<https://sysml.org/>



- Petri Nets: matematický a grafický model obsahující dva typy uzlů (místa a přechody), značení (marking) míst a pravidla pro změnu (firing)
 - **Místa** (Places) – vyjadřují podmínky a nesou tečky (tokens)
 - **Tečky** – reprezentují tok informace po síti
 - **Přechody** (Transitions) – vyjadřují události (změnu stavu), spuštěním (firing) přechodu nastane událost
 - **Značení** (Marking) – rozmístění teček po Petriho síti, reprezentuje stav



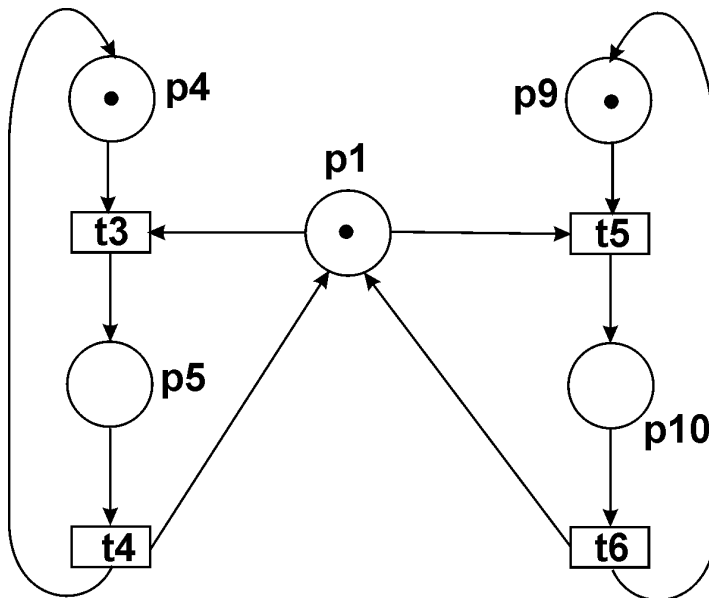


- **dualita:** dvě disjunktní množiny prvků:
 - *P-elements* (místa - *places: passive elements*, conditions, places, resources, waiting pools, channels etc.)
 - *T-elements* (přechody - *transitions: events, transitions, actions, executions of statements, transmissions of messages etc.*).
- **lokalita:** chování přechodu závisí výhradně na jeho lokalitě
- **souběžnost (concurrency):** přechody, které mají disjunktní lokalitu mohou nastat nezávisle na sobě
- **grafická reprezentace**
- **algebraická reprezentace**

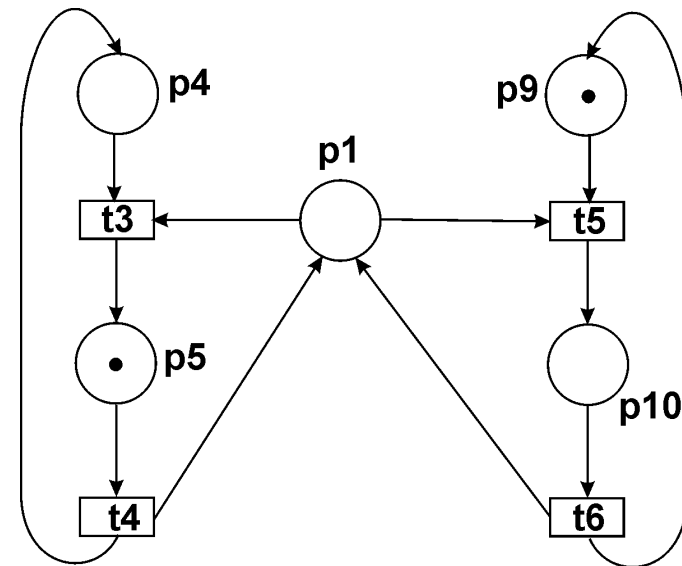
PN is more powerful model than FSM, but can be equivalent → use it only when necessary!

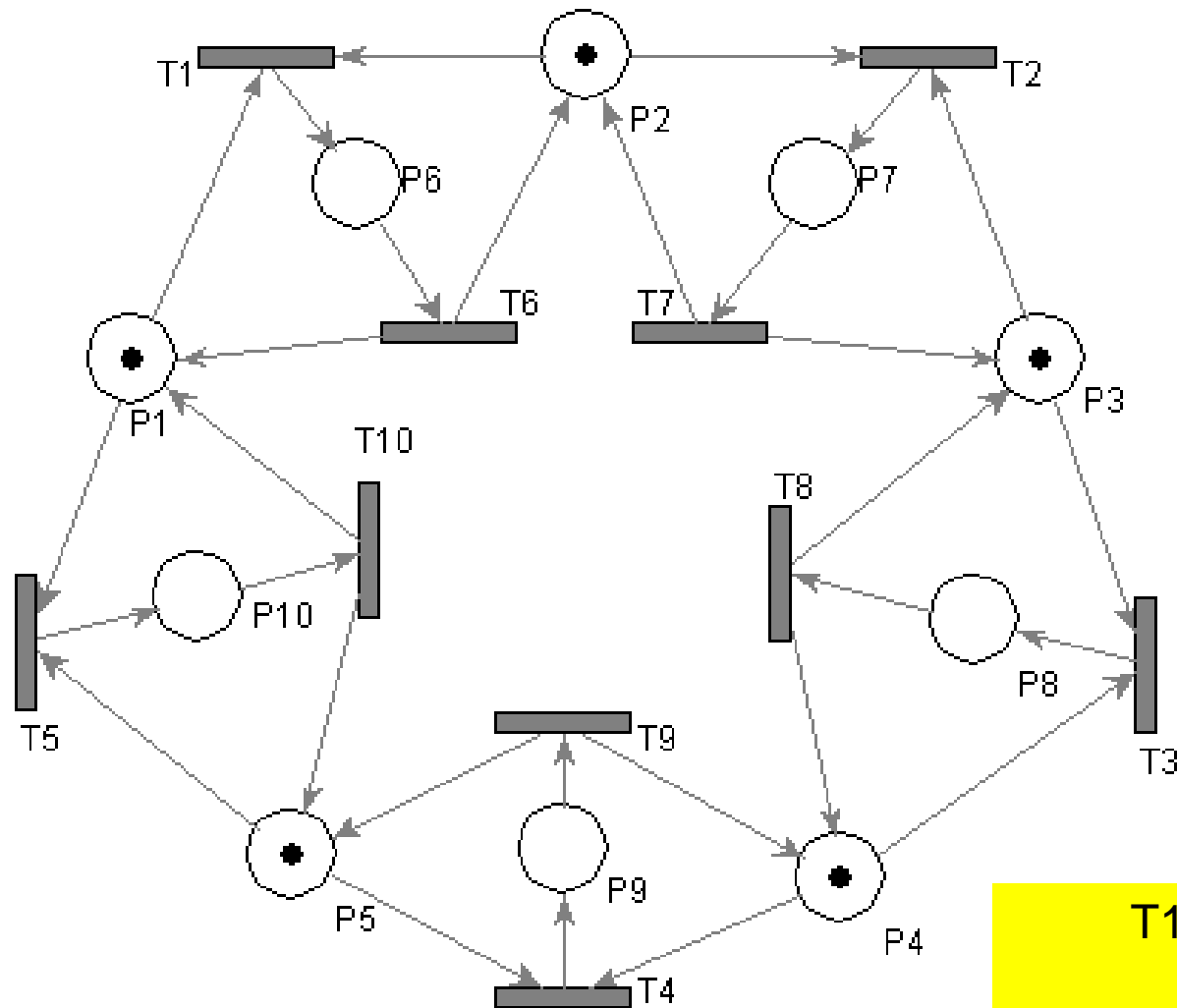


Mutual exclusion of places p5 and p10, transition t3 and t5 in conflict state where t3 and t5 are both enabled



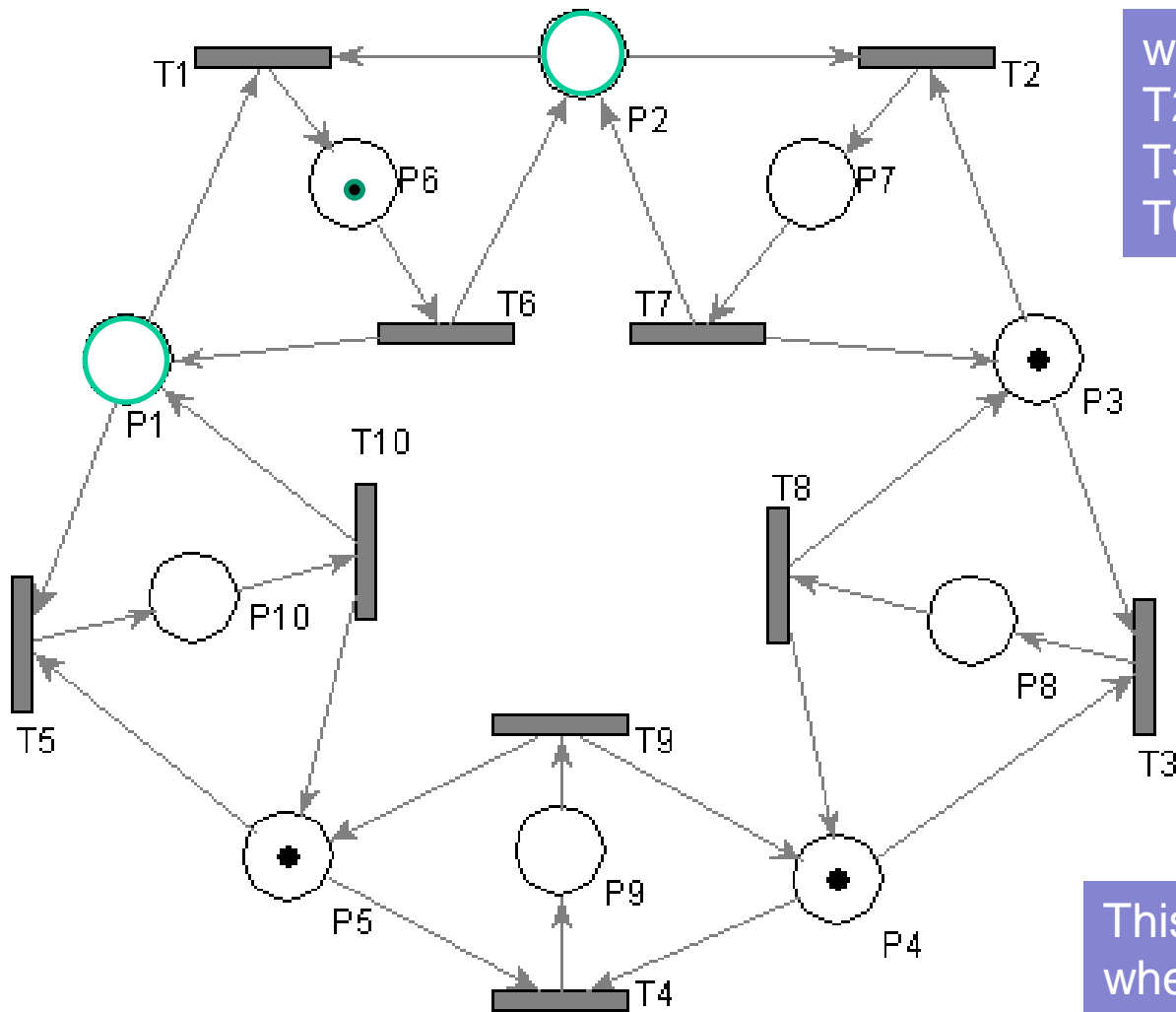
the state after t3 firing, where t5 is not enabled





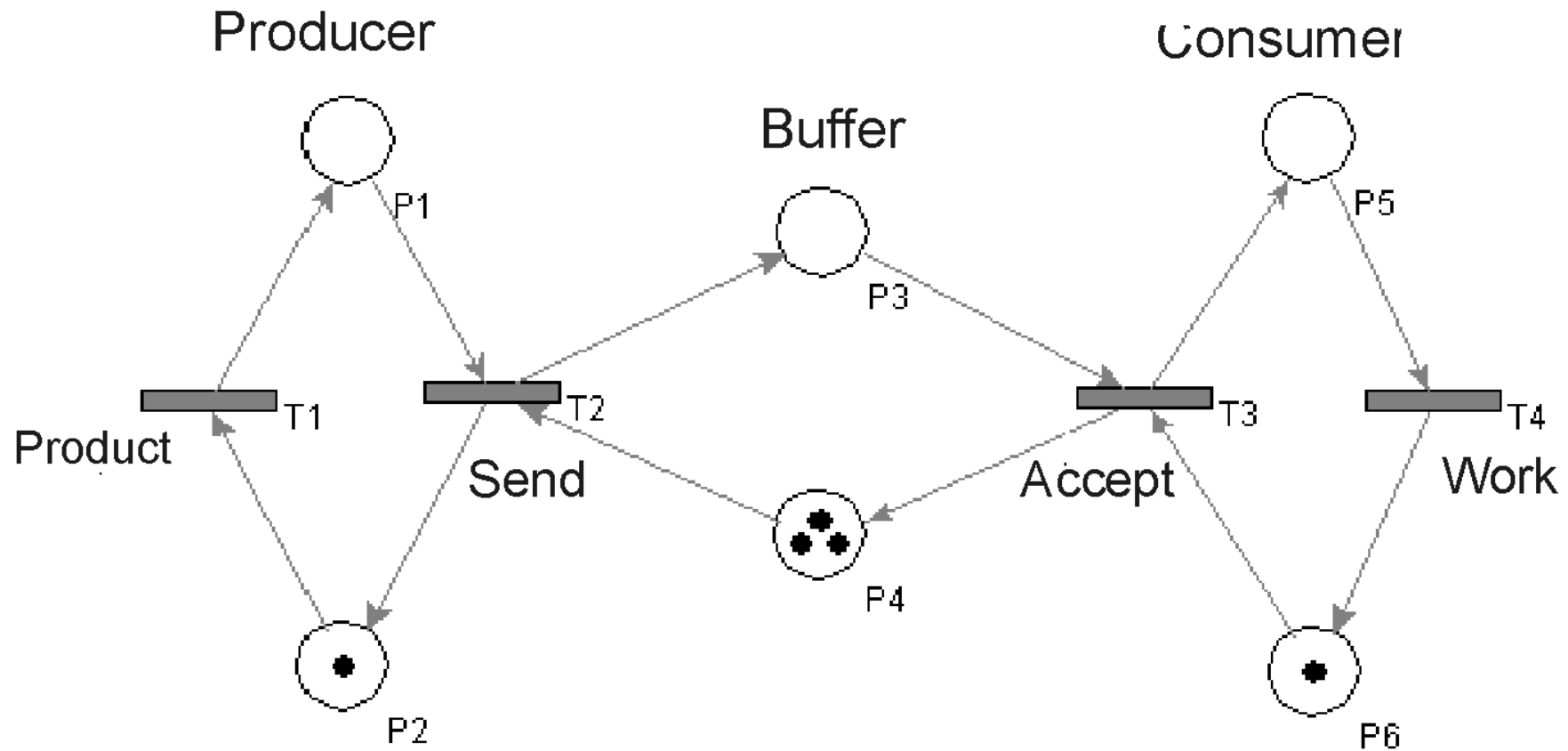
5 philosophers
each needs 2 forks to eat
5 fork places: P1 – P5
5 eat places: P6 – P10

T1, T2, T3, T4, and T5 are enabled
= all philosophers can start to eat
but not all transitions can fire concurrently



when T1 fires:
 T2 and T5 are not enabled
 T3 and T4 are enabled
 T6 and T3 or T4 can fire in parallel

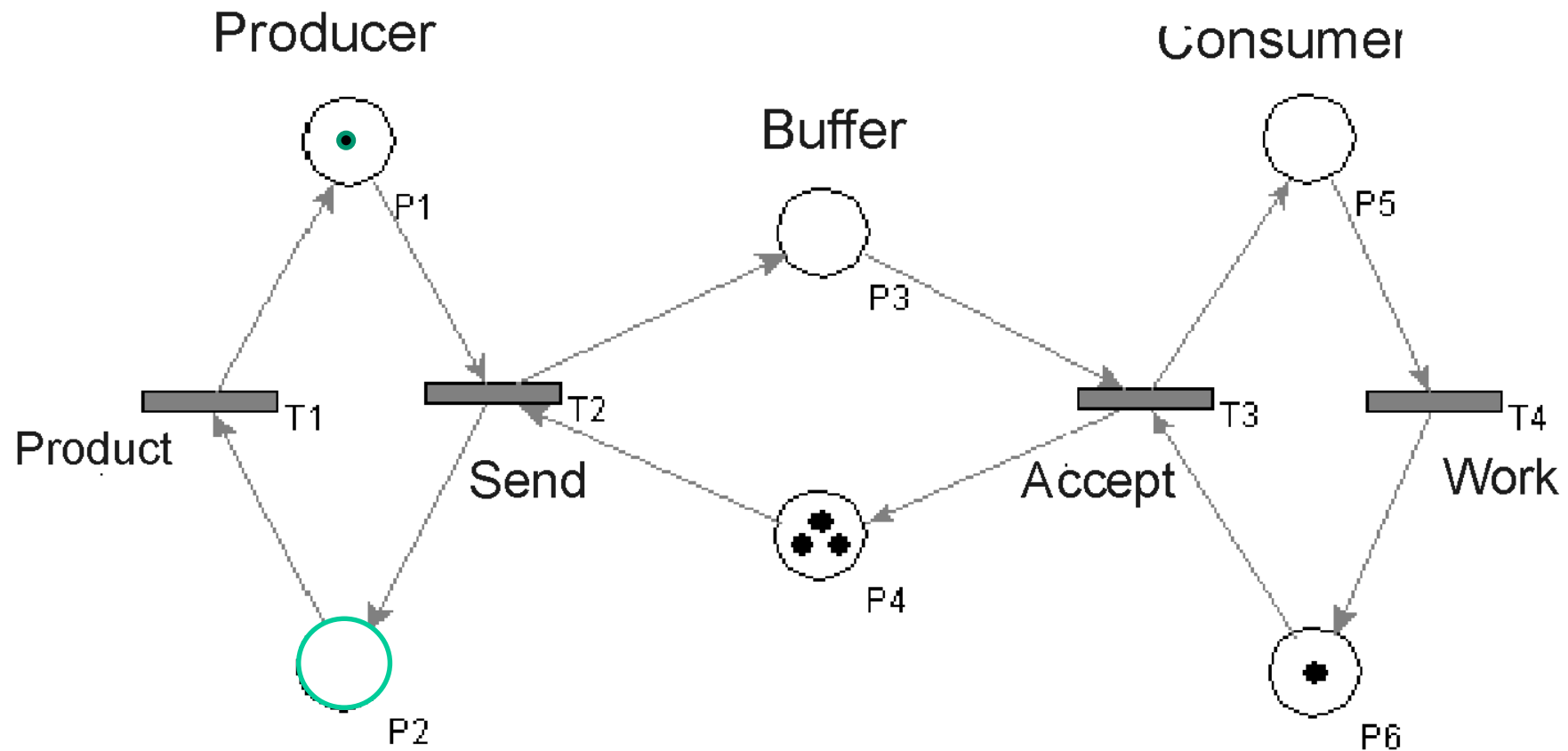
This model can overcome deadlock,
 when all 5 philosophers
 take left hand fork but right hand one
 is not available (or vice versa)



initial state:
only T1 is enabled

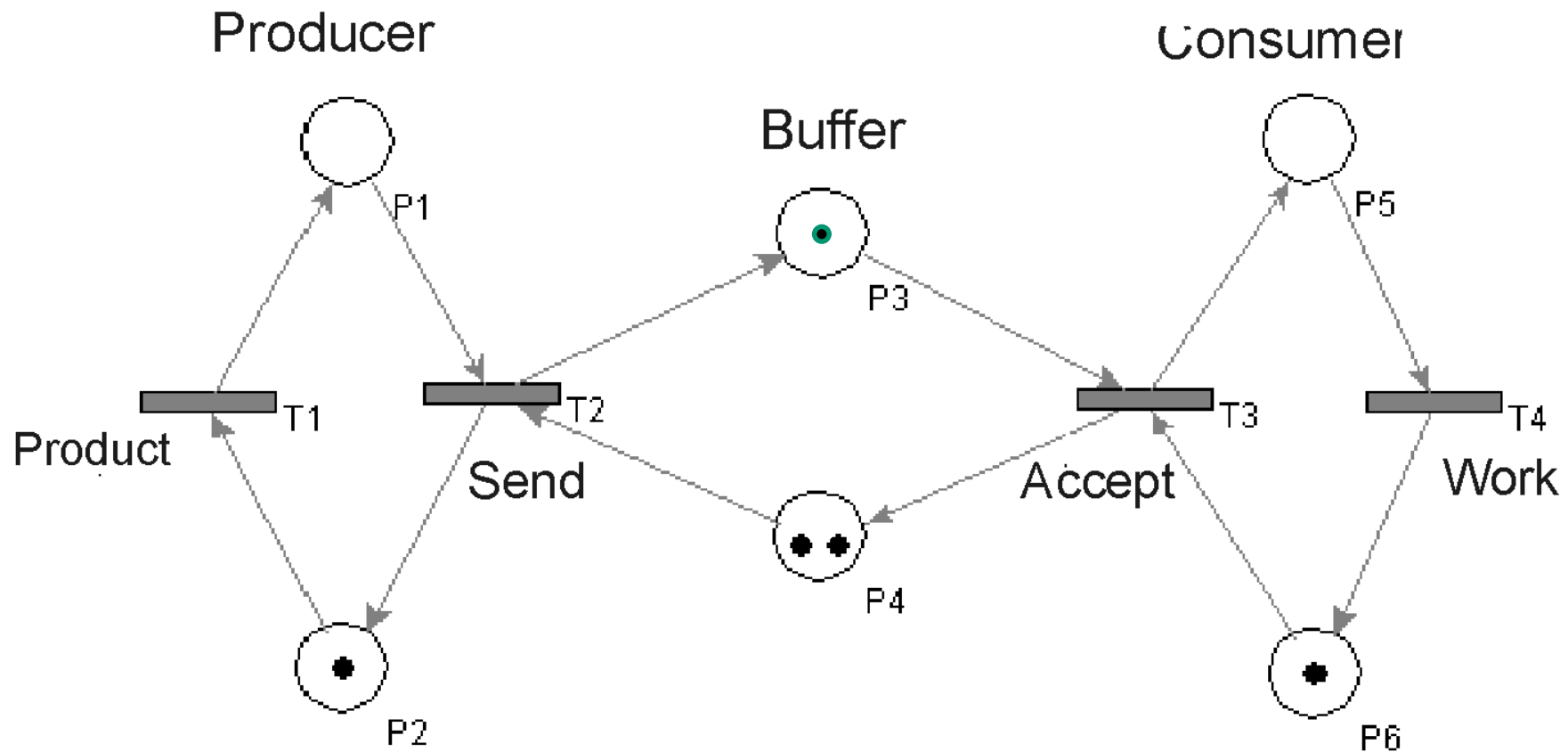


next state after firing T1:
now only T2 is enabled



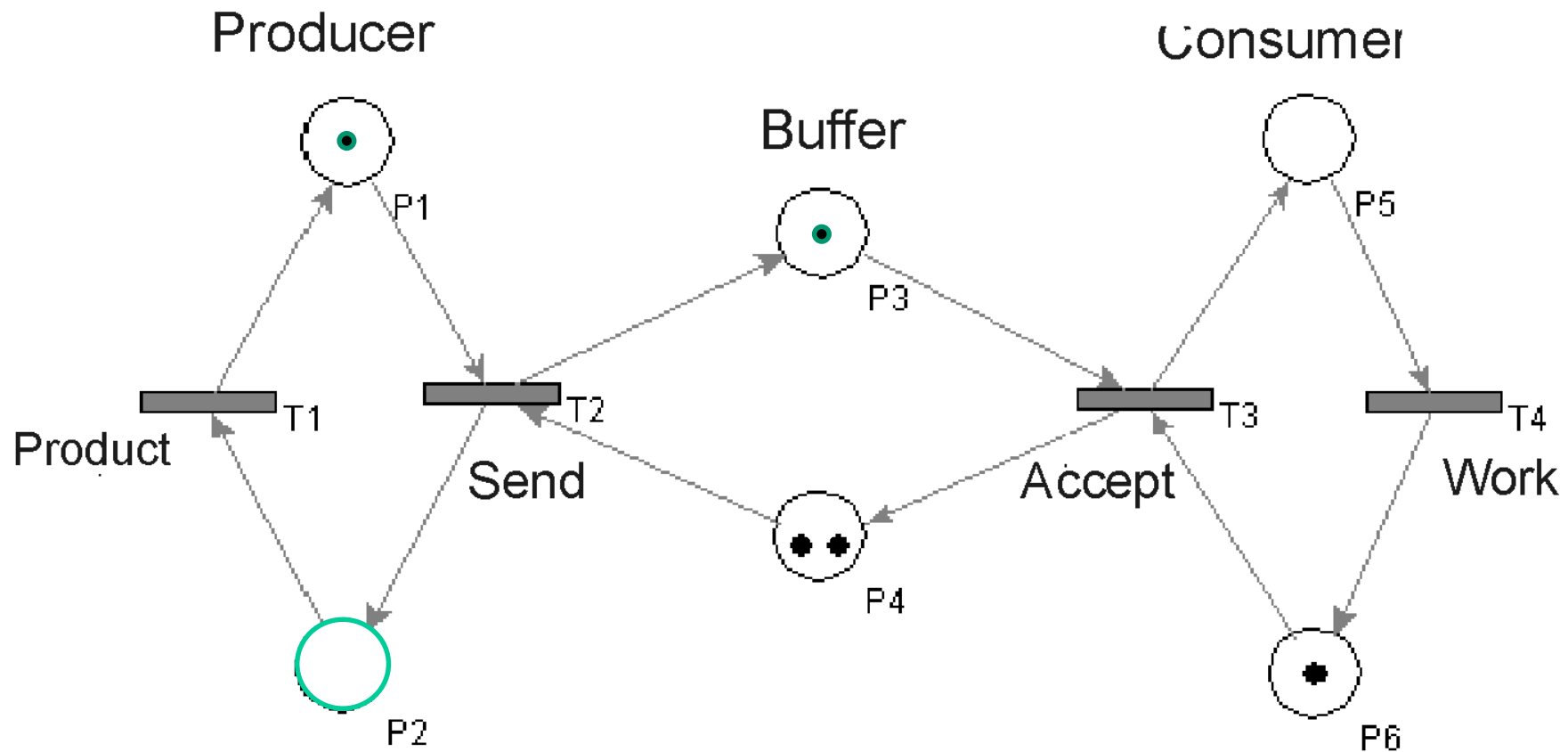


next state after firing T2:
only T1 and T3 are enabled



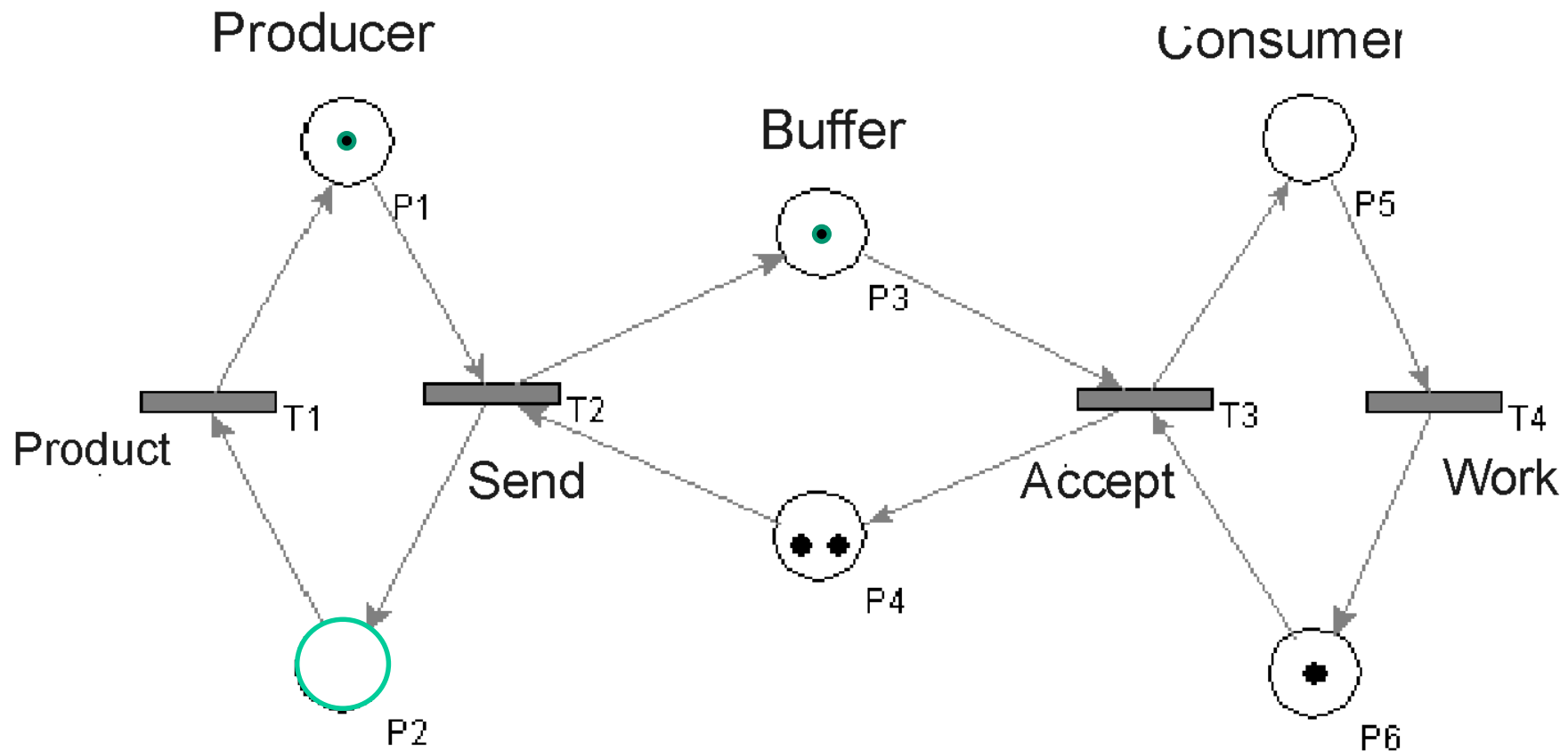


next state after firing T1:
only T2 and T3 are enabled





There are several possibilities, many states modelled by simple graphical model
it depends on capacity of buffer,
it is possible to model independent behavior of producer and consumer



Je třeba holistický pohled:

- modely ... hierarchické, podrobné,
- dedikované procesy a metodické postupy
- universální konstrukční principy (propojující modelovací principy s realitou vývoje a implementací)
- automatizovatelné kroky zajišťující rychlost návrhu