



## APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice - HIL Hlavní body přednášky

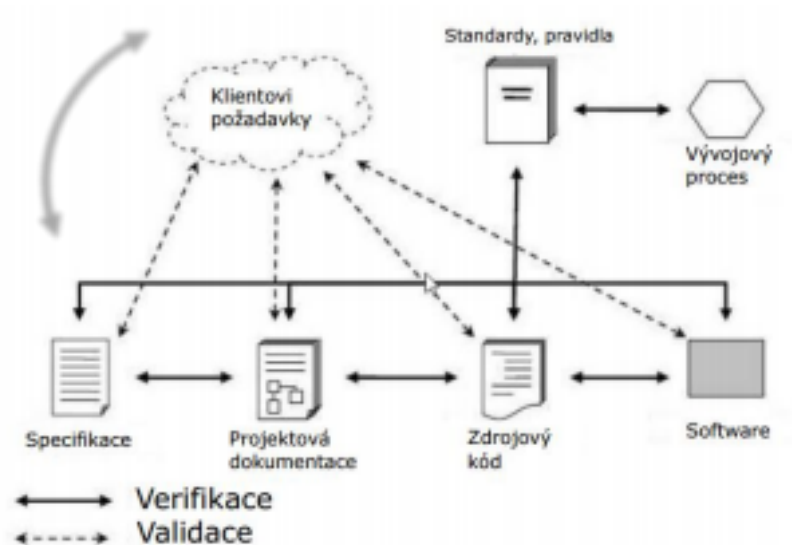
- ▶ Význam a cíle testování v elektronice
- ▶ Druhy testování
- ▶ Funkční (Black Box) testy
- ▶ Strukturální (White Box) testy (JTAG)
- ▶ Model Based Design a testování
- ▶ HIL (Hardware In The Loop) testování

# Význam a cíle testování v elektronice

**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice Testování

- ▶ Znamená porovnávat aktuální a očekávané chování.
- ▶ Znamená detekovat selhání (Failure).
- ▶ Je řízením rizik.
- ▶ Dává zpětnou vazbu vývojářům



Testování lze chápat jako **validaci** a **verifikaci** systému.

**Validace** – je kontrola, zda vyvíjený produkt splňuje požadavky dané zákazníkem

**Verifikace** – je kontrola, zda vyvíjený produkt odpovídá našemu návrhu (modelu). Mezi metody verifikace patří většina testování, kterého se neúčastní zákazník (unit testy, integrační testy,..)



## **APLIKACE – primární faktor definující požadavky na řízení**

### **Testování v elektronice Testování**

Klíčem k úspěšnému testování a **návrhu testů** obecně je detailní znalost **specifikace** systému a **systémových požadavků**.

Při testování elektronických systémů se využívá **modelování poruch**.

Je-li model poruch správně navržen a vyzkoušen, přisuzujeme mu věrohodnost a elektronický systémem, který tímto modelem otestujeme můžeme prohlásit za spolehlivý.

Podkladem pro **návrh modelu** poruch by měla být **specifikace testovaného systému**

a požadavky na očekávané chování systému ze strany zákazníka.

**Závada (fault)** - je následek pochybení člověka a je příčinou chyby

**Chyba (error)** – je stav systému, který může vést k selhání.

**Selhání (failure)** – je nesoulad mezi aktuálním a specifikovaným chování systému

## APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Testování

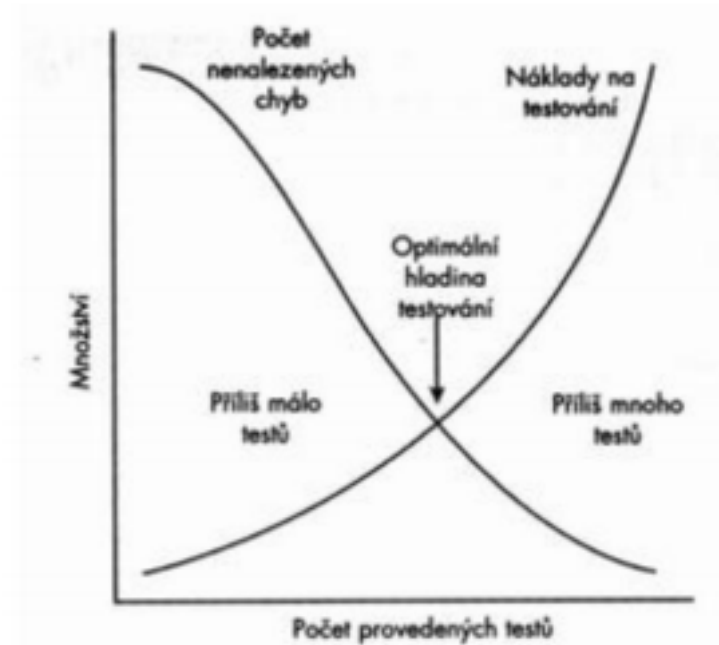
Návrh modelu poruch je vždy kompromisem

mezi mírou testovatelnosti systému (počet a typ provedených testů) a náklady na testování samotné.

**Pokrytí testy (test coverage)**

**Pokrytí testy (test coverage)**

- Pro konkrétní část specifikace by měl být navržen konkrétní test



- ▶ Produkt se považuje za spolehlivý, pokud je míra **pokrytí testy 100%**

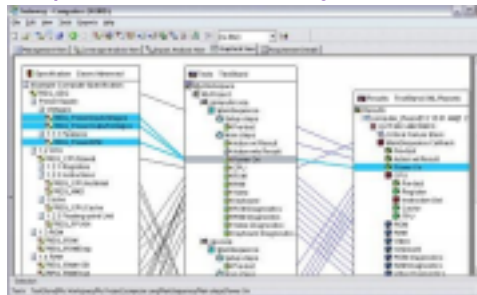
## APLIKACE – primární faktor definující požadavky na řízení

### Testování v elektronice Testování

#### Nástroje test management

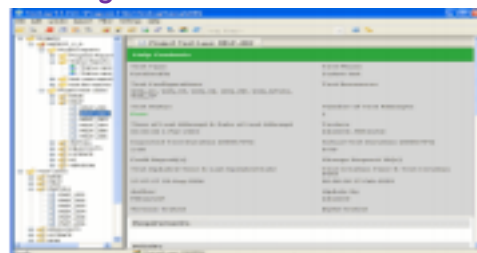
Jedním z možných způsobů jak sledovat fázi testování produktu je využití nástroje **Requirements Gateway** od spol. National Instruments. Dalším užitečným nástrojem je např. **TestLog**, určený pro správu testovacích případů.

#### *Requirements Gateway*



Ve spolupráci s SW TestStand, nástroj Requirements Gateway umožňuje import specifikace/í produktu a její provázání s jednotlivými testy *TestLog* a výsledky těchto testů.

Uživateli tak umožňuje sledovat vlastní proces a výsledky testování.



Dalším důležitým pojmem v oblasti testování je **testovací případ**, který popisuje konkrétní akce prováděné se systémem a jejich očekávané výsledky.

Testovací případ je tedy dokument, popisující určitou činnost, kterou je, na základě specifikace, třeba otestovat

# APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Testování

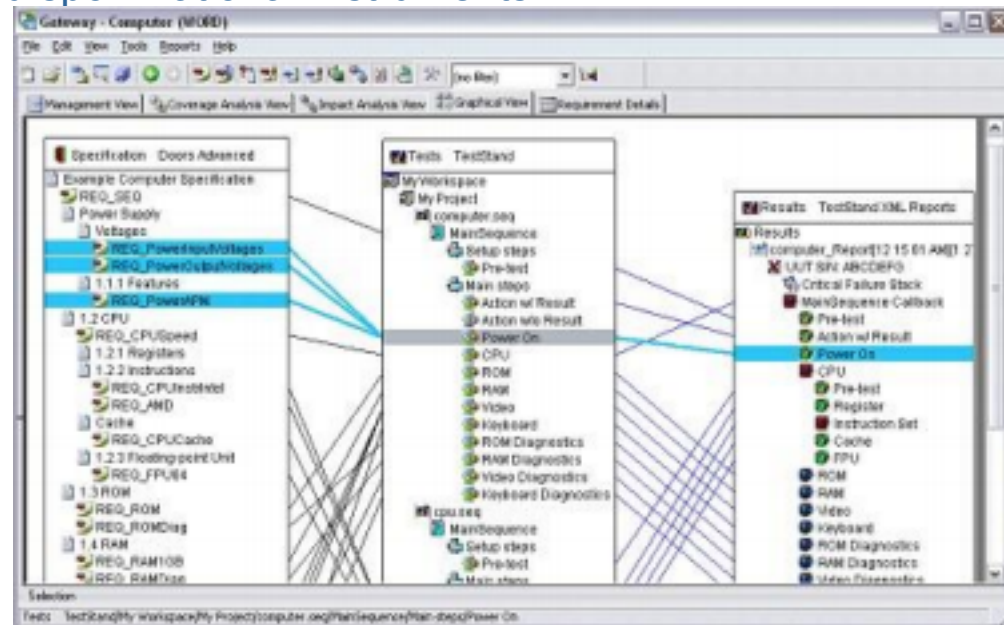
## Testovací případ (test case)

Jedním z možných způsobů jak sledovat fázi testování produktu je využití nástroje **Requirements Gateway** od spol. National Instruments.

Ve spojitosti s SW  
TestStand,

nástroj umožňuje import  
specifikace produktu a její  
provázání s jednotlivými  
testy a  
výsledky těchto testů.

Uživateli tak umožňuje  
sledovat proces a výsledky  
testování.





**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice Vývoj testování

### 1. Manuální testy

- Chyby způsobené testerem
- Vysoká náročnost z pohledu času i lidských zdrojů
- Vhodná pro malý počet testů a opakování

### 2. Skriptovací testování

- + navržené testy dle specifikace
- + automatizovatelnost spouštění
- + vhodné pro dlouhodobé a často opakované testování

### 3. Testování založené na modelu testované soustavy

- + jednoduchá správa testovacích případů



- + snížení nákladů a časové náročnosti
- + automatizovatelnost
- + teamová spolupráce na složitém zařízení



**APLIKACE – primární faktor definující požadavky na řízení**

## **Testování v elektronice Druhy testů**

### **1. Klasické pojetí testování**

- **Funkční (Black Box) testy**
- **Strukturální (White Box) testy**

### **2. Testování v uzavřené smyčce**

- **MIL (Model In the Loop)**
- **SIL (Software In the Loop)**
- **PIL (Processor In the Loop)**
- **HIL (Hardware In the Loop)**



**Testování v elektronice Funkční (Black Box)**

# testy



**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice Funkční (Black Box) testy

Při testu je simulováno provozní prostředí testované jednotky. Výchozím bodem pro návrh testu je testovací specifikace.

### Výhody

- ▶ Ověřuje systém jako celek
- ▶ Obtížná příprava testu a vlastní provedení testu



### Nevýhody

# automatizace

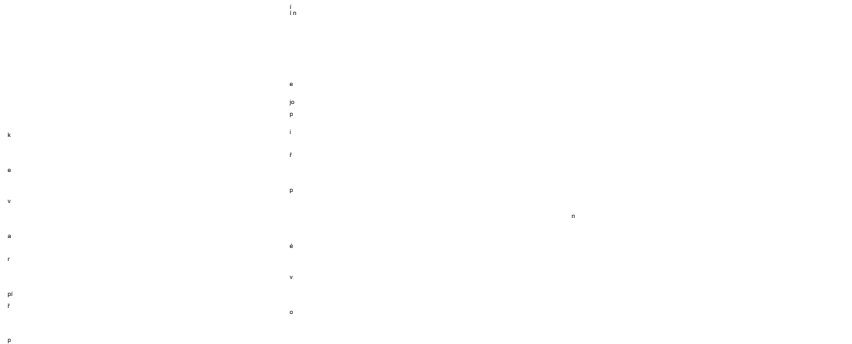
- ▶ Náročná příprava prostředí (Test Setup)
- ▶ Testy tohoto typu v laboratorních podmínkách vyžadují vyškolenou obsluhu



**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice Automatizace funkčních testů

Příklad zjednodušeného schématu automatizace pro funkční testování



Měřicí  
přístroje

Testovací

X

DMM

Zátěž

matice

řetězec

Sig.gen.

propojov  
ací



**APLIKACE – primární faktor definující požadavky na řízení**

**Testování v elektronice Příklad uspořádání testeru pro automatizaci**

**funkčních testů System pro testování řídicích bloků jaderných**

**elektráren**

Vana pro testované bloky 1.generace

Univerzální propojovací platforma

PXI šasí s měřicími kartami

Vana pro testované bloky 2.generace

Zdroje a zátěž pro výkonové testy



**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Testovací systém pro řídicí bloky jaderných  
elektráren**

Vana pro testované bloky 1.generace



**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Testovací systém pro řídicí bloky jaderných  
elektráren**

**Univerzální propojovací platforma se sdílenou sběrnicí čítající cca 150  
signálů + přepínací maticové karta**

Strana připojení



## Přizpůsobení signálových úrovní



Testování v elektronice

# Strukturální (White Box)

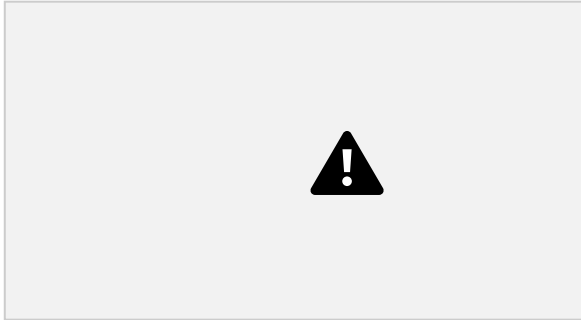
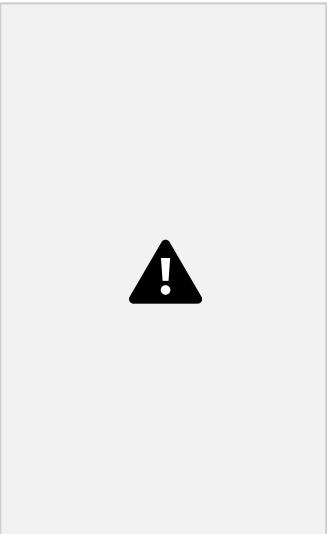
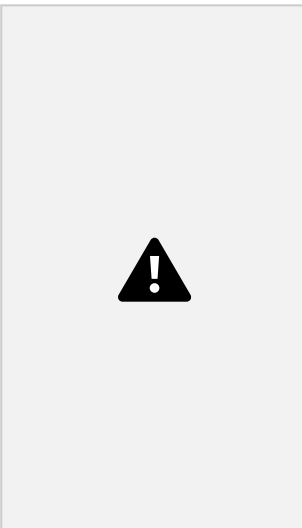
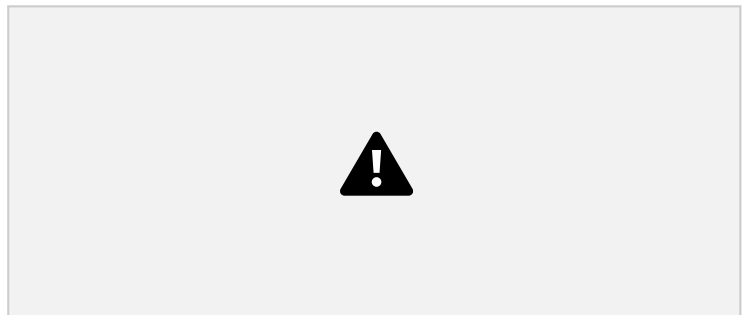
## testy JTAG



**APLIKACE – primární faktor definující požadavky na řízení**

**Testování v elektronice Strukturální (White Box) testy**

Známe vnitřní strukturu systému.



**JTAG**



**APLIKACE – primární faktor definující požadavky na řízení**

# Testování v elektronice JTAG standard

Joint Test Action Group (r. 1990)

## 1. Digital I/O IEEE Std. 1149.1

IEEE Standard Test Access Port and Boundary - Scan

## Architecture 2. Analog I/O IEEE Std. 1149.4

IEEE Standard for a Mixed-Signal Test Bus

## 3. AC – coupled digital pins IEEE 1149.6

IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks



**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice JTAG rozhraní

Čipy, navržené v souladu s IEEE 1149.1 jsou vybaveny dodatečnou logikou umožňující JTAG rozhraní provádění výše uvedených operací

## Interface je ve standardu označován jako Test Access Port (TAP)



**Interface obsahuje 5 signálů**

- TDI ... test data in
- TDO ... test data out
- TMS ... test mode select
- TCK ... test clock
- TRST ... low active reset (optional)

## JTAG chain



### APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Boundary Scan



- ▶ Po obvodu jádra čipu se u každého pinu nachází registr (tzn. „**Boundary Scan Register**“ (BSR))
- ▶ V případě spuštění aplikace („**normal operation mode**“) tento registr propojuje externí pin přímo s vnitřní logikou jádra čipu.

- ▶ Při spuštění „test mode“ odpojuje BSR logiku jádra a pin je ovládán přímo BSR registrem
- ▶ Přes BSR registr může pin ovládán jako výstup nebo čten jako vstup



APLIKACE – primární faktor definující

# požadavky na řízení Testování v elektronice Řídící a datový registr

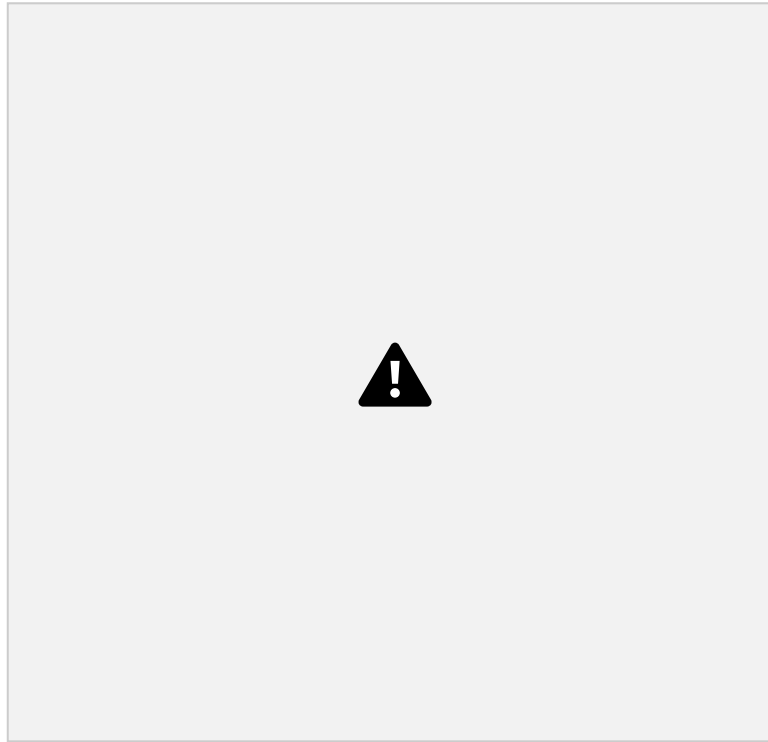


- ▶ „Bypass Register“ - „zkratka“ ze vstupu na výstup. Aktivací se vyřazuje průchod informace BSR registry a tím se zrychluje odezvu systému.
- ▶ Použití v řetězci čipů propojených JTAG rozhraním, kde přemostění čip není aktuálně objektem testování
- ▶ Instrukční registr vybírá příslušný datový registr pro čtení / zápis (viz výše)

Instrukční registr



**APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Stavový diagram TAP kontroléru**



<https://www.youtube.com/watch?v=PhaqHKyAvR4>



## **APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Druhy testovatelných poruch**



- ▶ Zkrat
- ▶ Rozpojená cesta
- ▶ Stuck at 1
- ▶ Stuck at 0



## APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice

### Propojení chipů na obecném schématu PCB

- ▶ Kaskáda čipů, kompatibilních se standardem IEEE 1149.1, propojená pěticí signálů: JTAG rozhraní tvořící Boundary Scan Chain (BSCH)



- ▶ PCB může obsahovat 1 i více BSCH
- ▶ BSCH poskytuje

vysokorychlostní  
sériovou sběrnici  
zpřístupňující  
zdroje pro testování  
PCB



## **APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Test propojení mezi JTAG chipy**

- ▶ Možnost testování několika propojení najednou









**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Testování součástek, které nemají zabudován  
JTAG**



- ▶ Pokud je izolovaný chip (cluster) vodivě propojen s jiným „JTAG“ chipem, testování je v omezené míře možné i v tomto případě.



**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Testování pamětí (SDRAM, DDR3, DDR4)**













**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Možnost programování přidružených Flash,  
EEPROM paměť**















**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Programování PLD (CLPD, FPGA)**

















**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice Debugging mikroprocesorové jednotky**



















## APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice JTAG – nástroje

- ▶ **XJAnalyser** – grafická aplikace, ovládání jednotlivých pinů
- ▶ **XJDeveloper** – test součástek mimo JTAG chain
- ▶ **XJRunner** – prostředí pro běh testů generovaných



z XJDeveloper modulu



[www.xjtag.com](http://www.xjtag.com)



**APLIKACE – primární faktor definující požadavky na řízení  
Testování v  
elektronice  
JTAG aplikace**

**Shrnutí**

- ▶ Nízká pořizovací cena
- ▶ Jediný nástroj
- ▶ Možnost ISP (In System Programming) ve všech cyklech výroby
- ▶ Využití pro malé série i masovou produkci
- ▶ Testování obtížně přístupných pinů (vysoký stupeň integrace, BGA ...)



## Testování v elektronice

**MBD / MBT**

# Model Based Design & Testing



## Testování v elektronice

### V-diagram modelově orientovaného vývoje v prostředí Matlab+Simulink

- **MIL** = Model In the Loop – Model ve smyčce
- **SIL** = Software In the Loop – Software ve smyčce (pozor může znamenat i Safety Integrity Level)
- **PIL** = Processor In the Loop – Procesor ve smyčce
- **HIL** = Hardware In the Loop – Hardware ve smyčce



## **APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Výhody MBD oproti klasickému vývojovému cyklu**

- ▶ Není třeba čekat na vývojový prototyp řízené soustavy – návrh řídicího systému může začít na základě systémové analýzy
- ▶ Odhalení chyby již v raném stádiu vývoje (MIL, SIL, PIL a HIL testy) ▶

Snadná integrace částí modelu, rozdělení práce na modelu v rámci týmu



**APLIKACE – primární faktor definující požadavky na řízení**

## **Testování v elektronice MIL (Model In the Loop)**

Jedná se obecně o techniku testování SW bloků proti sobě navzájem v prostředí, běhovém emulovaném SW frameworkem.

Běhové prostředí spuštěného testu pak zajišťuje vzájemné propojení modelů a předávání dat mezi nimi. Většinou umožňuje i budit modely vnějšími stimuly.

## Výhody / nevýhody

- ▶ Metody mohou, ale nemusí běžet v reálném čase (k běhu stačí běžné PC)
- ▶ Míra idealizace celé úlohy je značná (není zahrnut výpočetní výkon a četné nelinearity vyplývající z reálné aplikace)
- ▶ Vhodné pro testy, ověřující základní funkčnost testovaných algoritmů a zda SW splňuje specifikaci požadavků.



**APLIKACE – primární faktor definující požadavky na řízení**

### **Testování v elektronice SIL (Software In the Loop)**

V mnoha ohledech podobný MIL. Kód testované jednotky je vygenerován, zkompilován a přenesen zpět do simulačního prostředí.

Technika je vhodná zejména v aplikacích, kdy je kombinován automaticky

generovaný kód s přidaným, vlastnoručně psaným kódem.

Sada testovacích dat je většinou stejná jako při MIL testu

### Výhody / nevýhody

► Viz MIL.



## **APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice PIL (Processor In the Loop)**

Jde o typ testování, při kterém je testovaná část SW přenesena do reálného HW. Protistranu pak tvoří SW model běžící většinou ve standardním PC.

Data jsou mezi HW as SW modelem předávána většinou pomocí jedné ze standardních komunikačních sběrnic (dnes již převážně JTAG).

Simulace je řízena z PC, kde běží simulační model úlohy. Mohou zde být komunikována a zobrazována data o chování testovaného modelu.

### Výhody / nevýhody

► Simulovat lze v reálném čase



(ověření

výkonu HW a kvalitu regulace)

- ▶ Simulace mimo reálný čas umožňuje provedení mnoha testů, ověřující např. chyby zaokrouhlování (fixed point, floating point), odhalit problémy s optimalizací generovaného kódu compilerem, latence ISR, ADC,...



**APLIKACE – primární faktor definující požadavky na řízení**

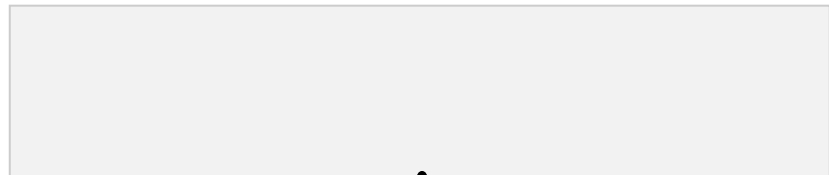
## **Testování v elektronice HIL Hardware In the Loop)**

Dalším krokem po PIL testu je logicky testování řídicího systému přímo v cílové aplikaci (řízené soustavě, motoru ...). Pokud je cena řízené soustavy vysoká a hrozí její poškození vlivem nevhodné regulace, nebo prototyp soustavy neexistuje, je použití HIL testů opodstatněné.

Model řízené soustavy je v tomto typu testu přenesen do HW, který zde přímo emuluje fyzické rozhraní mezi řídicí deskou a řízenou soustavou v reálném čase.

### **Výhody / nevýhody**

- ▶ Možnost simulovat i takové poruchy,



keré

bychom si v reálné aplikaci nemohli dovolit.

- ▶ Zhodnocení reálných výsledků kvality regulačního zásahu.
- ▶ Při testování lze údaje ze snímačů nastavit na konstantní hodnoty a při simulaci měnit pouze jeden parametr – odhalení mnoha skrytých chyb



**APLIKACE – primární faktor definující požadavky na řízení**

**Testování v elektronice Srovnání HIL a PIL**



JTAG Okruh využití simulačních metod typu PIL i HIL je velmi široký. Uplatňují se v letectví, armádě, automobilovém průmyslu apod., v podstatě všude tam, kde jde o složité řídicí úlohy na komplexních soustavách. Jejich velkým přínosem je schopnost odhalit mnoho skrytých problémů před tím, než je řídicí systém skutečně realizován v celém rozsahu a použit; to se pozitivně promítne v mnoha nezanedbatelných ohledech. Vytvořit zázemí pro tyto simulační metody sice nelze bez vynaložení velkého počátečního úsilí, to se však zpravidla v budoucnu mnohonásobně vrátí.



# HIL

## Hardware In the Loop testy



## APLIKACE – primární faktor definující požadavky na řízení

### Testování v elektronice - HIL HIL (Hardware In The Loop)

Četnost a komplexnost elektronických systémů v **automotive** stále narůstá. Elektronikou jsou postupně nahrazovány i některé mechanické systémy. Finanční a časové nároky na testování jsou proto značné. HIL testování umožňuje nahradit části systému matematickým modelem a testovat zařízení v reálném čase.

#### Výhody

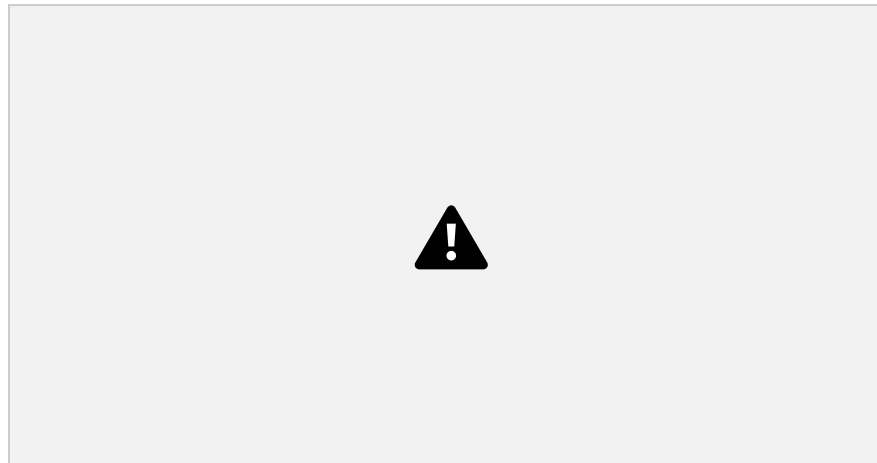
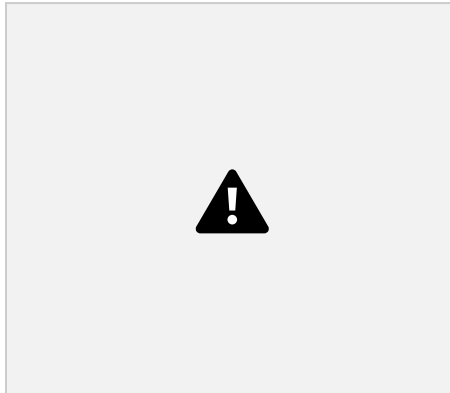
- ▶ Ověřuje funkce jednotlivých komponent ve funkčním celku – testuje se komunikace mezi nimi
- ▶ Odhalení chyb v raném stadiu vývoje (systémy jsou většinou vyvíjeny paralelně)



- ▶ Možnost simulovat i chybové stavy, které by na reálném zařízení nebylo možné připustit.



## **APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice - HIL HIL framework**





**APLIKACE – primární faktor definující požadavky na řízení**

## **Testování v elektronice - HIL Funkční celky obecné RT aplikace (HIL)**

- ▶ RT Stimulus – křivky popisující změny veličiny v čase
- ▶ Logování dat pro pozdější rozbor
- ▶ Možnost řízení a automatizace testu
- ▶ Zobrazování dat (minimálně poruchových stavů)
- ▶ Zpracování HW I/O
- ▶ Synchronizace mezi RT systémy

- ▶ Sdílení dat mezi RT systémy
- ▶ Škálování a kalibrace vstupních a výstupních veličin

Výše uvedené SW části tvoří holý základ (framework), nad kterým je teprve možné začít stavět HIL testovací aplikaci...



**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice - HIL Obecná skladba HIL systému

Real – time HW (PC, PXI, cRIO...)

Běžné PC (host)

LAN



Deterministická časová smyčka  
(zpracování I/O)

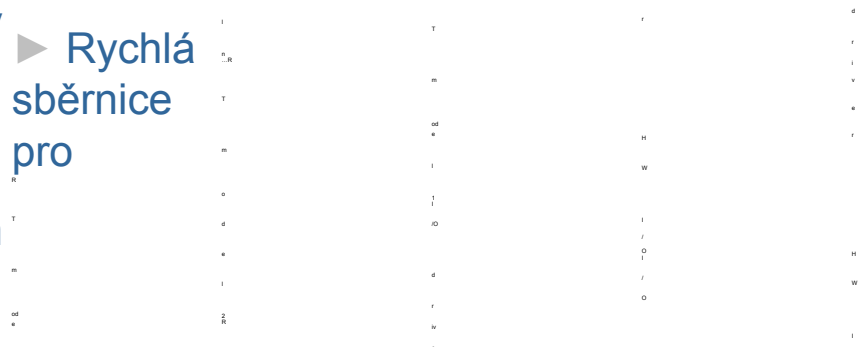
Real time SW (Dspace, Veristand, ...)

je výkonný počítač vybavený real-time operačním systémem

▶ Rychlá sběrnice pro



Základem



komunikaci s HW

► Ovládání HILu z PC

po LAN

Testovaný HW (DUT, SCU  
ECU)



**APLIKACE – primární faktor definující požadavky na řízení**

## Testování v elektronice - HIL NI Veristand Engine

- Základem je PCL (Primary Control Loop) – běží s nejvyšší prioritou a řídí časování celého systému
- Ostatní smyčky běží paralelně s prioritou stejnou nebo nižší než PCL
- Komunikaci mezi smyčkami zajišťují vnitřní vyrovnávací fronty (FIFO)



- V každé iteraci jsou přečteny vstupy, provedeny výpočetní operace a zapsány výstupy.
- Custom Device umožňuje vložit vlastní HW zařízení nebo SW modul



## APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice - HIL Navýšení výpočetního výkonu



**(FPGA)** Využití FPGA ve VeriStandu spočívá ve dvou krocích:

1. Příprava modelu
2. Vytvoření Custom Device



**Model**

**LabVIEW / Simulink**

**FPGA karta PXI, cRIO + RT Linux**





# APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice - HIL Postup vytvoření real-time aplikace v NI Veristand

## 1. System Explorer

1. Import zkompilovaných modelů (Simulink, Labview)
2. Import Custom Devices (FPGA)
3. Přidání NI-DAQ zařízení (analogové / digitální vstupy)
4. Vytvoření kanálů s možností přepočtu vstupní, výstupní veličiny pomocí transformační rovnice
5. Mapování kanálů = vzájemné propojení kanálů modelů a kanálů HW

## 2. Stimulus Profile

1. Vytvoření real-time profilů pro stimulaci modelů / HW
2. Logování dat

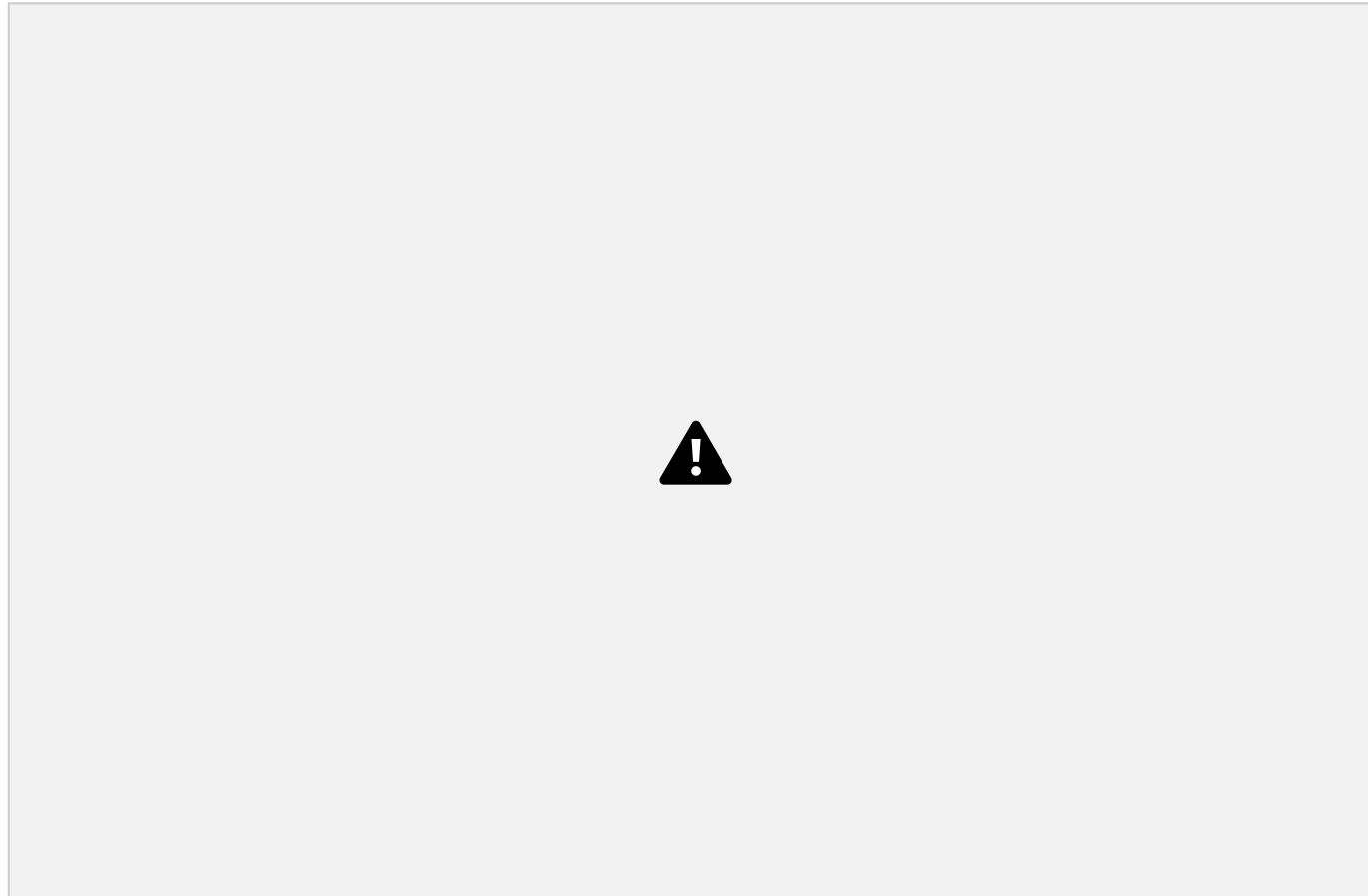
## 3. User GUI

1. Použití knihovny grafických prvků k vytvoření ovládaní běhu aplikace, zobrazování dat



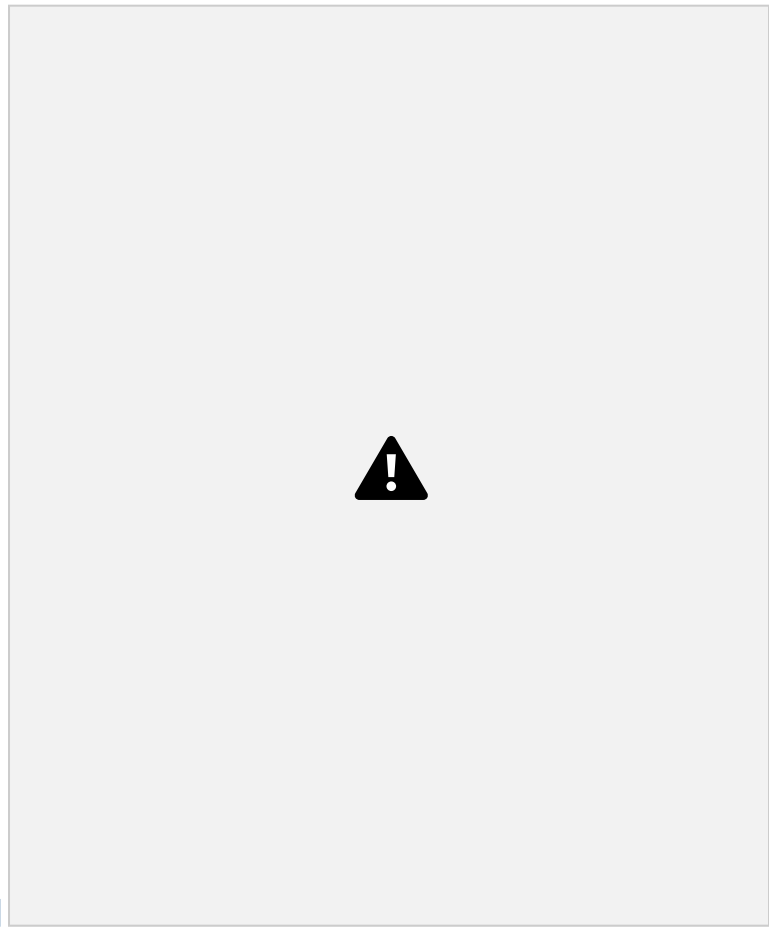
# APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice - HIL

## NI Veristand: System Exporer



**APLIKACE – primární faktor definující požadavky na řízení**

**Testování v elektronice - HIL NI Veristand: Vstupní a výstupní**



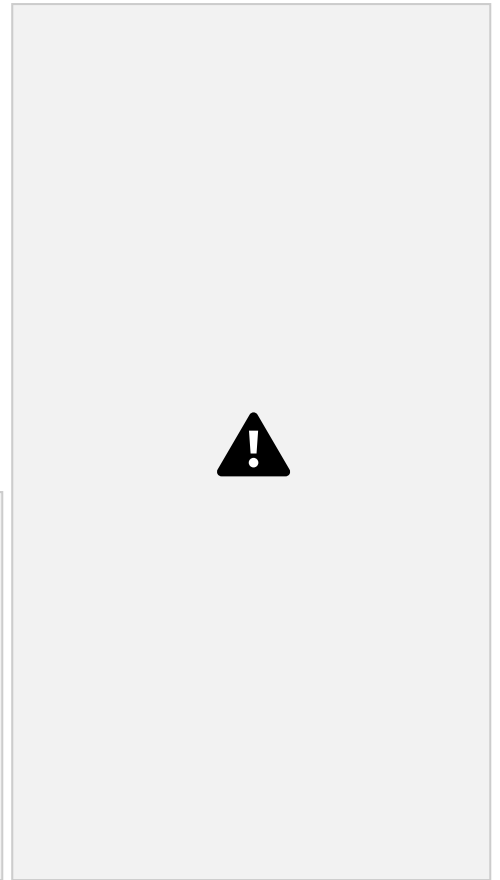
porty importovaného modelu



**APLIKACE – primární faktor definující požadavky na řízení**  
**Testování v elektronice - HIL NI Veristand: Mapování kanálů**



**APLIKACE – primární faktor definující požadavky na řízení  
Testování v elektronice - HIL NI Veristand: Stimulus Profile  
Editor**





## **APLIKACE – primární faktor definující požadavky na řízení Testování v elektronice Použité zdroje**

- ▶ Simulace Processor In the Loop a Hardware In the Loop
  - ▶ [https://automa.cz/cz/casopis-clanky/simulace-processor-in-the-loop-a-hardware-in-the-loop-2007\\_05\\_34311\\_2055/](https://automa.cz/cz/casopis-clanky/simulace-processor-in-the-loop-a-hardware-in-the-loop-2007_05_34311_2055/)
- ▶ Aplikace testování založeného na modelu testovaného systému na HIL platformě
  - ▶ [https://dSPACE5.zcu.cz/bitstream/11025/23640/1/DP\\_Bohman.pdf](https://dSPACE5.zcu.cz/bitstream/11025/23640/1/DP_Bohman.pdf)
- ▶ Modifikace nástroje TaSysTest pro běh v prostředí Real-Time
  - ▶ [https://wiki.control.fel.cvut.cz/mediawiki/images/1/16/Dp\\_2016\\_zak\\_pavel.pdf](https://wiki.control.fel.cvut.cz/mediawiki/images/1/16/Dp_2016_zak_pavel.pdf)

▶ Processor-In-Loop Simulation: Embedded Software Verification & Validation In Model Based Development

▶ <https://www.design-reuse.com/articles/42548/embedded-software-verification-validation-in-model-based-development.html>

▶ Processor-in-the-Loop Simulations Applied to the Design and Evaluation of a Satellite Attitude Control

▶ <https://www.intechopen.com/books/computational-and-numerical-simulations/processor-in-the-loop-simulations-applied-to-the-design-and-evaluation-of-a-satellite-attitude-contr>



