

# Funkční verifikace číslicových systémů: Verifikace TIMERu

**Autor:** Pavel Kratochvíl (xkrato61)

**Dátum:** 24.4.2025

## 1 Implementácia referenčného modelu

Referenčný (*golden*) model bol implementovaný v súbore `golden_model.svh`, pomocou ktorého bola overená funkčnosť TIMERu.

Porovnávaním hodnôt v scorebode bola odhalená chyba v implementácii TIMERu (`timer_fvs.vhd`), ktorá spočívala v zlých odpovediach pre určité hodnoty vstupu ADDRESS. Podľa špecifikácie by hodnota výstupu RESPONSE mala byť určená nasledovne:

- IDLE - ak je REQUEST NONE; inak
- ERROR - ak je REQUEST RESERVED; inak
- OOR - ak je hodnota ADDRESS > 0x14; inak
- UNALIGNED - ak je hodnota ADDRESS nezarovnaná; inak
- ACK

Táto logika bola v pôvodnom kóde implementovaná zle a preto som ju opravil aby odpovedala špecifikácii (obr. 4).

```
92      -- corrected
93      bus_resp_d <=
94      CP_RSP_IDLE      when unsigned(REQUEST) = CP_REQ_NONE      else
95      CP_RSP_ERROR     when unsigned(REQUEST) = CP_REQ_RESERVED else
96      CP_RSP_OOR       when unsigned(
97      |               |   ADDRESS(TIMER_ADDR_SPACE_BITS-1 downto 0)
98      |               |   ) > unsigned(TIMER_CYCLE_H)             else
99      CP_RSP_UNALIGNED when ADDRESS(1 downto 0) /= "00"          else
100     CP_RSP_ACK;
101
```

Obr. 1: Screenshot opravenej logiky určenia hodnoty signálu RESPONSE na požiadavky READ a WRITE.

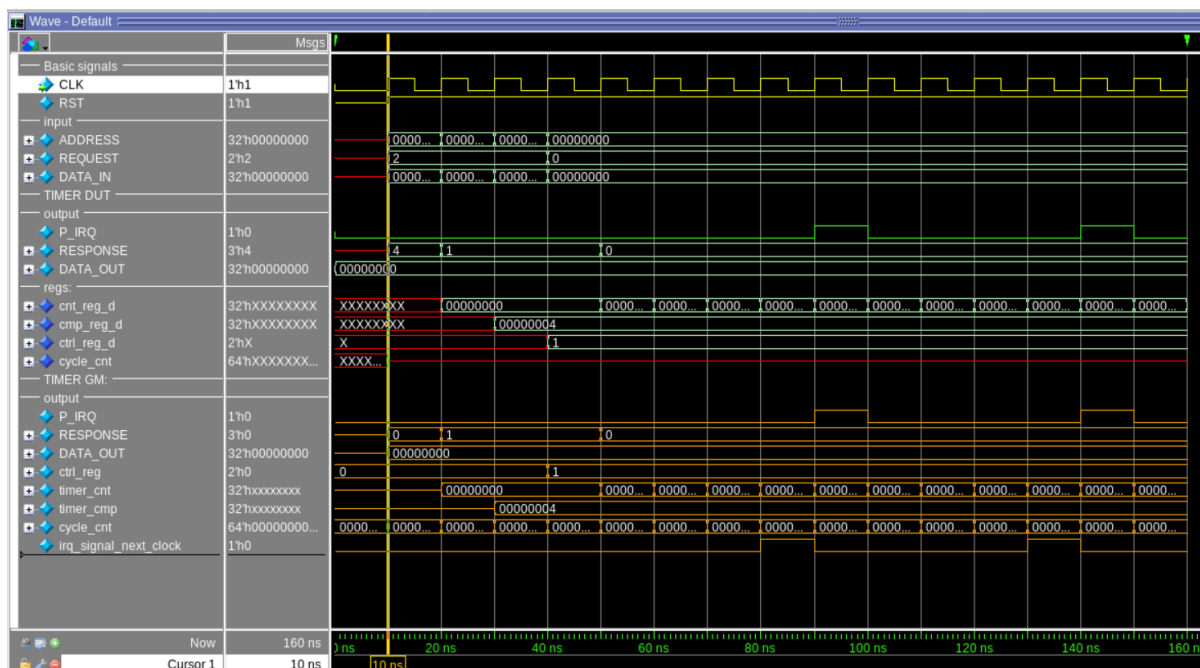
Podľa zadania bola overená funkčnosť základného testu `timer_t_test` aj bez resetovacej sekvencie. Tu som objavil ďalšiu potencionálnu chybu, kedy timer DUT inicializuje hodnoty vnútorných registrov až po uvedení signálu reset do aktívnej hodnoty. Teda bez iniciálnej reset sekvencie sú hodnoty nedefinované (obr. 2). Korektné správanie je implementované v referenčnom modeli, kedy sa po štarte inicializujú všetky hodnoty registrov na nulové hodnoty.

## 2 Náhodné testy a constraints randomizovaných premenných

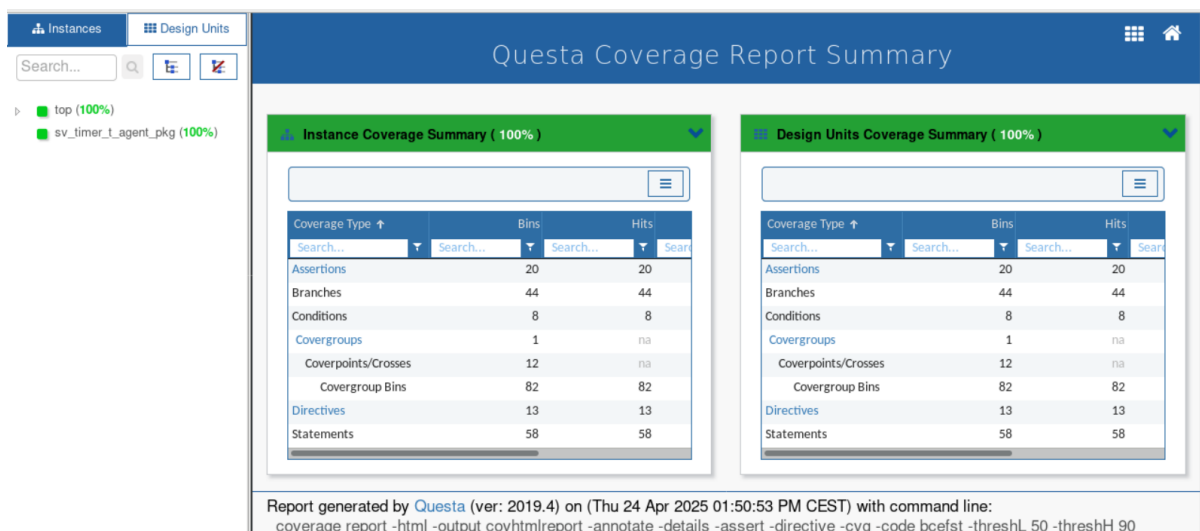
Boli implementované dva pseudo-náhodné testy. Prvý (`pseudo_random_test`) bol vypracovaný s použitím constraints na randomizované premenné podľa zadania z druhého cvičenia (trieda `extended_timer_t_transaction` v súbore `transaction.svh`). Ten však sám nedosahoval 100% pokrytie. Preto som implementoval pseudonáhodnú sekvenciu `new_sequence`, kde som obmedzil pomocou constraints v triede `extended1_timer_t_transaction` hodnoty RST, REQUEST, ADDRESS, DATA\_IN tak, aby boli viac zamerané iba na platné hodnoty a sústredili sa tak primárne na zmeny módu TIMERu a na čítanie/zápis do registru CNT.

## 3 Implementácia scenárov funkčného pokrytia

Podľa inštrukcii z tretieho cvičenia som doplnil priame testy a pseudo-náhodné testy (opísané v predošlej časti). Bolo úspešne dosiahnuté 100% pokrytie kódu, ako aj 100% funkčné pokrytie. Na dosiahnutie plného pokrytia som použil v oboch pseudo-náhodných testoch 10000 transakcií (`TRANSACTION_COUNT`).



Obr. 2: Demonštrácia behu základného testu na TIMER DUT a referenčnom modeli (TIMER GM) bez iniciálnej reset sekvencie. Hodnoty registrov DUT sa stávajú definovanými až v momente, kedy sa do nich zapíše hodnota pomocou WRITE requestu (ku zápisom dochádza v prvých 3 taktoch).



Obr. 3: Snímok obrazovky vygenerovaného reportu (make coverage). Bolo dosiahnuté 100% funkčné pokrytie a 100% kódu DUT.

## 4 Implementácia formálnych tvrdení

Formálne tvrdenia boli vypracované podľa zadania zo štvrtého cvičenia. Všetky asserts a covers boli doplnené do súboru `abv_timer.sv`, kde som nadefinoval jednotlivé properties a následne som implementoval assert/cover pre všetky z nich s náležitými chybovými hláškami (pomocou funkcie `'uvm_error()`). Výsledky je možné vidieť na obrázkoch 5, 6 a 7.

Coverage Summary By Instance ( 100% )						
Instance ↑	Assertions	Branches	Conditions	Directives	Statements	Total
Total	100%	100%	100%	100%	100%	100%
HDL_DUT_U	100%	100%	100%	100%	100%	100%

Recursive Hierarchical Coverage Details ( 100% )						
Coverage Type ↑	Bins	Hits	Misses	Coverage		
Search...	Search...	Search...	Search...	Search...	Search...	Search...
Assertions	20	20	0	100%		
Branches	44	44	0	100%		
Conditions	8	8	0	100%		
Directives	13	13	0	100%		
Statements	58	58	0	100%		














Obr. 4: Snímok obrazovky vygenerovaného reportu (make coverage) zachycujúci detaily pokrytia assertions, branches, conditions, directives a statements.

Assertions					
Name	Assertion Type	Language	Enable	Failure Count	Pass Count
/top/HDL_DUT_U/abv_timer_module/a_ResetRegCheck	Concurrent	SVA	on	0	455
/top/HDL_DUT_U/abv_timer_module/a_ResetCycleCntRegCheck	Concurrent	SVA	on	0	455
/top/HDL_DUT_U/abv_timer_module/a_addrOORResponseCheck	Concurrent	SVA	on	0	3798
/top/HDL_DUT_U/abv_timer_module/a_signalsKnownInactiveResetCheck	Concurrent	SVA	on	0	19599
/top/HDL_DUT_U/abv_timer_module/a_prUnknownDataRead	Concurrent	SVA	on	0	4780
/top/HDL_DUT_U/abv_timer_module/a_prUnknownDataWrite	Concurrent	SVA	on	0	4921
/top/HDL_DUT_U/abv_timer_module/a_prReadWriteOOR	Concurrent	SVA	on	0	3798
/top/HDL_DUT_U/abv_timer_module/a_prReadWriteUnaligned	Concurrent	SVA	on	0	237
/top/HDL_DUT_U/abv_timer_module/a_checkWriteReadSameAddr	Concurrent	SVA	on	0	281
/top/HDL_DUT_U/abv_timer_module/a_ackAfterCorrectAddr	Concurrent	SVA	on	0	5242
/top/HDL_DUT_U/abv_timer_module/a_idleResponseToNonReq	Concurrent	SVA	on	0	7364
/top/HDL_DUT_U/abv_timer_module/a_errorResponseToResReq	Concurrent	SVA	on	0	2947
/top/HDL_DUT_U/abv_timer_module/a_noWaitResponse	Concurrent	SVA	on	0	19599
/top/HDL_DUT_U/abv_timer_module/a_irqAfterCmpCntMatch	Concurrent	SVA	on	0	882
/top/HDL_DUT_U/abv_timer_module/a_clearCntAfterCmpCntMatchAutoRestart	Concurrent	SVA	on	0	514
/top/HDL_DUT_U/abv_timer_module/a_incCntAfterCmpCntMatchContinuous	Concurrent	SVA	on	0	110
/top/HDL_DUT_U/abv_timer_module/a_clearCntDisAfterCmpCntMatchOneShot	Concurrent	SVA	on	0	81
/top/HDL_DUT_U/abv_timer_module/a_cycleLRead	Concurrent	SVA	on	0	9
/top/HDL_DUT_U/abv_timer_module/a_cycleHRead	Concurrent	SVA	on	0	16
/top/HDL_DUT_U/abv_timer_module/a_cycleCntZeroDuringReset	Concurrent	SVA	on	0	455
/uvm_pkg/uvm_reg_map/do_write/#ublk#215181159#1731/immed__1735	Immediate	SVA	on	0	0
/uvm_pkg/uvm_reg_map/do_read/#ublk#215181159#1771/immed__1775	Immediate	SVA	on	0	0

Obr. 5: Snímok výsledkov Assertions zo spojených výsledkov všetkých testov (final.ucdb).

INST Vsv_timer_t_agent_pkg::timer_t_coverage::FunctionalCoverage	100.00%	100	100.00%	✓
CVP modes	100.00%	100	100.00%	✓
CVP request	100.00%	100	100.00%	✓
CVP reset_inactive	100.00%	100	100.00%	✓
CVP cv4	100.00%	100	100.00%	✓
CVP addresses	100.00%	100	100.00%	✓
CVP irq	100.00%	100	100.00%	✓
CVP cv9	100.00%	100	100.00%	✓
CVP all_modes_transitions	100.00%	100	100.00%	✓
CROSS cv6	100.00%	100	100.00%	✓
CROSS cv7	100.00%	100	100.00%	✓
CROSS cv10	100.00%	100	100.00%	✓
CROSS cv12	100.00%	100	100.00%	✓

Obr. 6: Snímok výsledkov Cover Groups zo spojených výsledkov všetkých testov (final.ucdb).

Cover Directives											
Name	Language	Enabled	Log	Count	AtLeast	Limit	Weight	Cmplt %	Cmplt graph	Included	
▲ /top/HDL_DUT_U/abv_timer_module/c_prReadWriteOOR	SVA	✓	Off	3798	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_prReadWriteUnaligned	SVA	✓	Off	237	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_checkWriteReadSameAddr	SVA	✓	Off	281	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_ackAfterCorrectAddr	SVA	✓	Off	5242	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_idleResponseToNonReq	SVA	✓	Off	7364	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_errorResponseToResReq	SVA	✓	Off	2947	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_irqAfterCmpCntMatch	SVA	✓	Off	882	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_clearCntAfterCmpCntMatchAutoRestart	SVA	✓	Off	514	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_incCntAfterCmpCntMatchContinuous	SVA	✓	Off	110	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_clearCntDisAfterCmpCntMatchOneShot	SVA	✓	Off	81	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_cycleLRead	SVA	✓	Off	9	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_cycleHRead	SVA	✓	Off	16	1	Unlim...	1	100%		✓	
▲ /top/HDL_DUT_U/abv_timer_module/c_cycleCntZeroDuringReset	SVA	✓	Off	455	1	Unlim...	1	100%		✓	

Obr. 7: Snímok výsledkov Cover Directives zo spojených výsledkov všetkých testov (final.ucdb).