Contract Specification of Pipelined Designs

Alexander Kamkin
Institute for System Programming of RAS
kamkin@ispras.ru

Pipelined Designs

- Operations consist of several stages which are overlapped in execution
- Pipeline organization reduces average execution time per operation
- Pipeline organization introduces additional problems and new sources of errors

Functional Testing

- Functional testing of hardware models consumes 70-80% of the design efforts
- Requirements on thoroughness of hardware models testing are very strong
- Testing can’t be done manually due to the size and complexity of designs
- Formal specifications can be used for testbench automation

MIPS R4000 Errata

- Datapath bugs 6.5%
- Control logic bugs 93.5%

R. Ho, C. Yang, M. Horowitz, and D. Dill.
“Architecture Validation for Processors”

Suggested Approach

- Contract specifications in the form of preconditions and postconditions describe functionality of pipe stages
- Temporal binding mechanism combines specifications of separated stages into a co-operative specification

Contract Specification of Pipeline

- V – set of context variables
- $I \cup O \subseteq V$ – input and output parameters
- $X \cup \{\tau\}$ – set of stimuli
- $Z \cup \{\varepsilon\}$ – set of stages
- ρ – function of operation composition

Stimulus

- $in \subseteq I$ – set of input parameters of stimulus
- $use \subseteq V \backslash O$ – set of variables used by stimulus
- pre – precondition of stimulus

Clock Stimulus (τ)

- $in_\tau = \varnothing$
- $use_\tau = \varnothing$
- $pre_\tau \equiv true$

Stage

- out $\subseteq$ O – output parameters of stage
- use $\subseteq$ V\O – set of variables used by stage
- def $\subseteq$ V\I – set of variables defined by stage
- γ – guard condition of stage
- post – postcondition of stage

Empty Stage (ε)

- $out_{\varepsilon} = \varnothing$
- $use_{\varepsilon} = \varnothing$
- $def_{\varepsilon} = \varnothing$
- $\gamma_{\varepsilon} \equiv true$
- $post_{\varepsilon} \equiv true$

Operation Composition

- ρ: $(X \cup \{\tau\}) \rightarrow (Z \cup \{\varepsilon\})^{L}$ – function of operation composition
- $\rho(\tau) = (\varepsilon, \ldots, \varepsilon)$ – operation of the clock stimulus consists of empty stages

Control State

- Stimulus processing state is a pair (x, s)
  - x – stimulus being processed
  - s – stage of stimulus processing
- Control state is a set of stimuli processing states $\{(x_i, s_i)\}_{i=1,n}$
  - Initial control state is the empty set of stimuli processing states

Stimulus Processing

- Enabled(π) – set of enabled stimuli in state π

  $\{ (x, s) \in \pi \mid \text{guard of } \rho_s(x) \text{ is true } \}$

- Locked(π) – set of locked stimuli in state π

  $\{ (x, s) \in \pi \mid \text{guard of } \rho_s(x) \text{ is false } \}$

Pipeline Shift Operator

- $\circ : (X \cup \{\tau\}) \times State \rightarrow State$ – pipeline shift operator
- $(x \circ \pi)$ is the union of the following sets:
  - Locked(π)
  - $\{ (x, l+1) \mid (x, l) \in Enabled(\pi) \wedge l < L \}$
  - $\{ (x, 1) \}$

Temporal Binding Operator

- θ: State $\rightarrow$ Power(Z) – temporal binding operator
- $\theta(\pi)$ is the following set of stages:

  $\{ \rho_l(x) \mid (x, l) \in \text{Enabled}(\pi) \} \setminus \{ \varepsilon \}$

EFSM Interpretation

- S = State – states are control states
- Y = Power(Z) – reactions are sets of stages
- $\pi' = (x \circ \pi)$ – final state of transition
- $y = \theta(\pi')$ – reaction of transition

Semantics of Context Updating

- Test oracle of stimulus x in control state π:

$$\Pi_{z \in \theta(x \circ \pi)} post_z(Use, Def)$$

Tool Support

- The approach was integrated into the CTESK test development tool from the UniTESK toolkit
- Pipeline shift operator and temporal binding operator are implemented as library functions

Case Study

- Translation Lookaside Buffer (TLB):
  - Data address translation
  - Instruction address translation
  - Reading entry
  - Writing entry
  - Probing entry

Statistics

- Implementation
  - ~8 KLOC (Verilog HDL)
  - ~60 inputs and outputs
- Specification
  - ~100 requirements
  - ~2.5 KLOC (extension of C language)
- Errors
  - ~10 errors were found including critical ones

Conclusion

- Approach is suitable for testbench automation
- Approach is integrated into the CTESK test development tool
- Approach has been successfully applied to several units of the industrial microprocessor

Contacts

- Institute for System Programming of RAS
  http://www.ispras.ru
- UniTESK Test Development Technology
  http://www.unitesk.com
- Alexander Kamkin
  kamkin@ispras.ru

Thank You!

Questions?