

STEP - Automated Synthesis of On-Chip-Bus Protocol Transducers

Masahiro Fujita
VLSI Design and Education Center (VDEC)
University of Tokyo

VLSI Design and Education Center (VDEC) in the University of Tokyo has created a new tool called STEP which stands for SynThesizing Environment for Protocol transducer. The tool automatically generates RTL description of transducers between two on-chip protocols. STEP can generate minimum latency protocol transducers from the automaton based protocol definitions and is capable of handling complicated on-chip interfaces including non blocking, burst mode, out-of-order, and others. Evaluation results show that it takes only a couple of minutes to generate protocol transducers for the full set of OCP.

1 Incompatibility of Communication Protocols in Modern SoC Design

With increasing complexity of the circuits on a single chip, IP-based design methodology is attracting attention to shorten the design periods. By utilising existing designs for new designs, the design period is expected to be much shortened. In IP-based designs, one of the most important issues is the connectabilities between IPs. Since IPs usually have their own interfaces, they cannot communicate with one another if they use different protocols. In other words, IPs which can communicate with a given IP may be limited due to interface mismatch.

In actual designs, designers usually insert protocol transducers (also called wrappers or bridges) between IPs with incompatible protocols, which consume additional designers' time. As a result, the advantage of IP-based design is reduced. To resolve the problem, automatic synthesis of protocol transducers is an attractive solution. It, however, is not at all straightforward, as the state-of-the-art protocols such as OCP have various complicated functionalities such as non-blocking (pipelined) and out-of-order transactions. Unfortunately, the synthesis methods proposed so far are hard to deal with such advanced features in complicated protocols, because the synthesis algorithms used in the previous methods cannot deal with these features. Here we introduce an automatic transducer synthesis tool, STEP, which is capable of handling protocols with the advanced fea-

tures mentioned above.

2 Synthesis of Practical Protocol Transducers

2.1 Protocol Modeling

In STEP, the definitions of on-chip bus protocols are assumed to be given in a form of finite state machine (FSM) represented in XML format. If new bus protocols must be processed, users can simply give their protocol definitions in FSM, and STEP can generate protocol transducers between those protocols.

Communication protocols in STEP are modeled as sets of sequences. A sequence corresponds to one action in a protocol such as Single Read, Burst Write etc. It is modeled using a single automaton (for blocking protocols), or two automata (for non-blocking or out-of-order protocols). Correspondences between the sequences in two bus protocols are assumed to be given by the user.

2.2 Synthesis Process Divided per Sequence

The basic idea in our approach is to divide the description of whole protocol into sets of sequences. While a single automaton has been often used for a description of a protocol in previous works, modeling a complicated protocol with advanced features may result in a very large automaton for the complete specification. So in STEP, sequence level transducer synthesis for each pair of corresponding sequences is applied first to generate partial transducers. Then the whole transducer is constructed from the set of the previously generated partial transducers. The overall flow of the synthesis process in STEP is shown in Figure1.

The sequence level synthesis employs automaton level synthesis. Given a pair of sequences used in two protocols, STEP performs a depth-first search to identify valid pair of corresponding states in the two sequences. A naive depth first search may generate almost all pairs of states in the two protocols. A valid pair of states is defined as the states which does not cause the transducer to output un-received data.

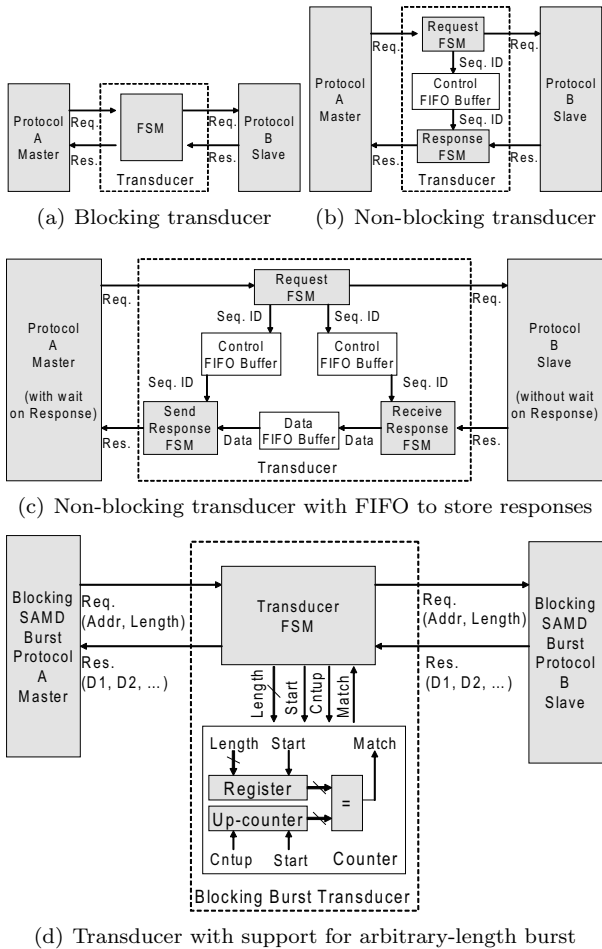


Figure 2: Supported transducer configurations in STEP

Among valid mappings between states, the mapping which minimizes the number of cycles required for completing the sequence is selected to synthesize the sequence level transducer. For protocols whose FSMs have loops, each loop is packed into one “super state” and mapping between states are searched in a hierarchical manner.

STEP supports four types of transducer configurations. For blocking protocols, the generated single FSM itself functions as the transducer between two protocols (Figure2(a)). For non-blocking protocols, a transducer with two FSMs synchronized through a FIFO which handle requests and responses separately (Figure2(b)) is synthesized. For configurations with a wait signal available only on the master side, transducers with a FIFO to store pending responses from the slave are synthesized (Figure2(c)). The sequence level synthesis is also capable of handling protocols with arbitrary-length burst mode operations by performing multi-body automaton level synthesis including an external circuit containing burst-length counter (Figure2(d)).

The FSM for the whole transducer is constructed by merging corresponding FSMs for sequence level transducers by sharing the initial and the final state in the transaction among all the sequence level transducers.

3 STEP in Action

STEP is implemented as a GUI application which runs both on Windows/Linux with more than 10,000 lines of C++ codes. In the following, the flow for synthesizing protocol transducers is demonstrated.

3.1 Loading Protocol Definitions

The first step of operations in STEP is to load target protocol definitions. Users can load protocol definitions for the bus master and slave by clicking “Load Master” button (A-1) and “Load Slave” button (A-2), respectively. After the protocol definitions are successfully loaded, list of available sequences will be shown in “MasterSequence” (A-10) and “SlaveSequence” (A-11).

3.2 Selecting a Structure for the Transducer

After loading protocol definitions, users specify which structure template to be used for the converter to be generated (Blocking, Non-blocking and Non-blocking with FIFO). Users select one of the supported structures from “ConstructType” list (A-3), and click “TypeSelect” button to fix the selection. “Hint” button gives brief descriptions for the structures.

3.3 Synthesizing Sequence Level Transducers

The sequence level synthesis can be started by selecting a sequence from each of MasterSequences list (A-10) and SlaveSequences list (A-11), and clicking SequenceLevelSynthesis button (A-6). Users are expected to synthesize all sequence level transducers, corresponding to sequences to be supported in the final whole transducer.

3.4 Synthesizing Whole Transducer

After synthesizing all sequence level transducers, users can construct the whole transducer, by clicking ConstructWholeTransducer button (A-7). After the synthesis is complete, report files shown in Figure4 are generated.

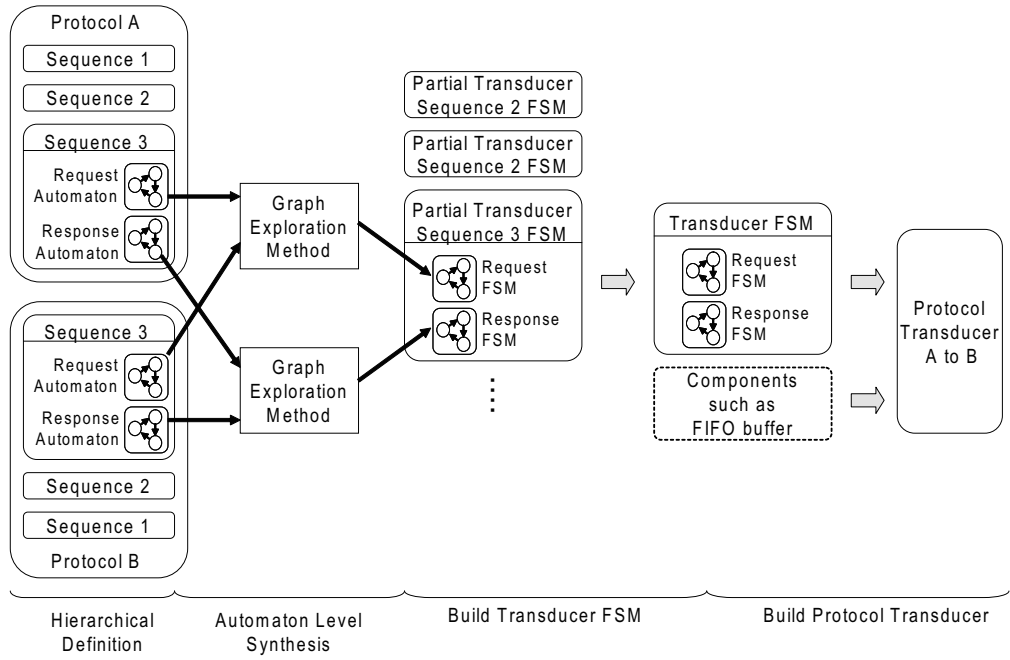


Figure 1: Protocol transducer synthesis flow in STEP

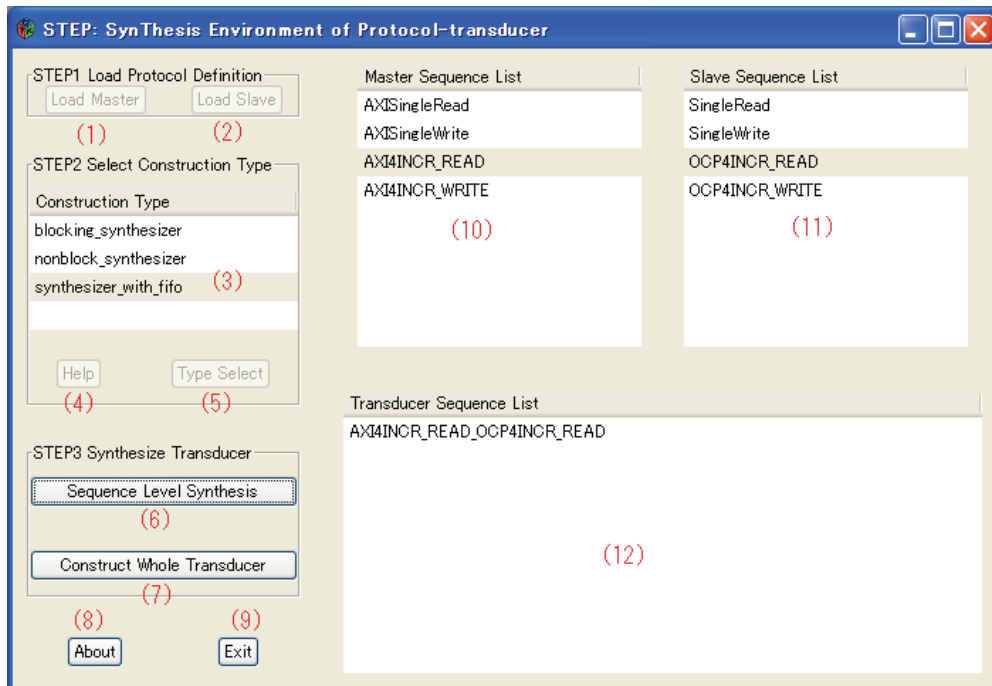


Figure 3: GUI console of STEP

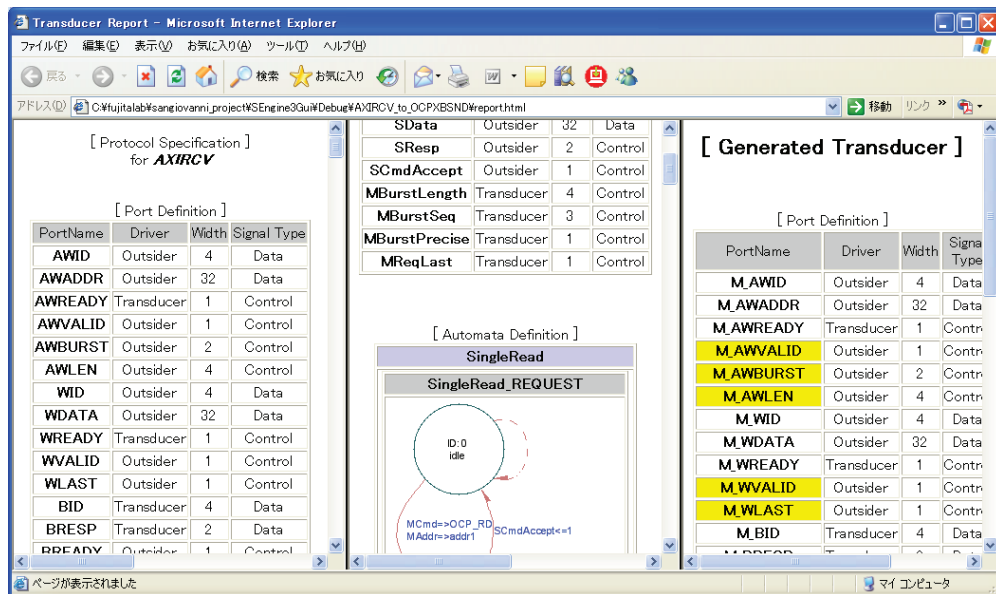


Figure 4: Synthesis reports from STEP. Each panel from the left corresponds respectively to reports on the master protocol definition, the slave protocol definition and the generated transducer.

Table 1: Resource usage of the generated transducers and run-time for the synthesis

Type	#of Seq.	States /Edges	Time [s]	Gates (ISE)	Area [μm^2]
AXI→OCV 1-4w FIX	6	64 /132	0.625	49804	196626
AXI→OCV 1-16w FIX	10	488 /946	2.234	261938	778804
OCV→AXI 1-16w VAR	2	14 /27	0.812	4682	32414

3.5 Examples of the Application

STEP is confirmed to be able to generate RTL descriptions of protocol transducers in Verilog-HDL between OCV and AMBA-AXI under several configurations of burst lengths and master/slave modes. Their functionality is confirmed through intensive RTL simulations. The designs generated by STEP are synthesized with XST in Xilinx ISE 9.1, and Cadence BuildGates. Resource usages of the synthesized designs are shown in Table1.

4 Summary

IP-based design methodology for shortening design periods often faces problems on incompatibilities of communication protocols between IPs. While protocol transducers for the connection of IPs with incompatible protocols has usually been designed manually, the cost for such manual designs reduces the advan-

tage of IP-based design.

Automatic synthesis of protocol transducers is an attractive solution. The methodologies proposed so far, however, are hard to handle the state-of-the-art protocols with advanced features such as OCV. STEP, which synthesizes protocol transducer designs from descriptions on sets of sequences used in the protocols, is capable of synthesizing protocol transducers between complicated protocols. The key techniques in STEP are division of synthesis process per transaction and multi-body synthesis including external circuits for storing burst count and/or other variables used in the protocol.

STEP will be offered shortly to members as part of valuable infrastructure from the OCV-IP web site.