*Christos P Sotiriou and Tiziano Villa*

# 1 Assignment 1 – Multiple FSMs to PTnet Conversion

As discussed in the Lectures, Morris Mano presents, in his Digital Design book, a very interesting, but rather complex to understand, Edge-Triggered Flip-Flop (FF) implementation. The specific FF design is illustrated in Figure 1. The clock signal comes in the middle section, whereas the D input comes in as the bottom input.

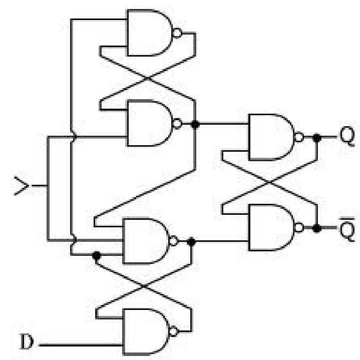


*Figure 1 - Edge-Triggered Flip-Flop Gate-Level Implementation*

A **key property of this FF implementation is that the inputs of the output Set-Reset (SR) Latch can never both be simultaneously zero**, *i.e.* the 00 value cannot occur, and further on, this implementation **will never allow at the SR Latch inputs a 00→11 transition to occur**, which would trigger the outputs Q and Q' to go metastable.

The goal of Assignment 1 is to perform a rigorous Formal Analysis and Verification of this FF design. A total set of four steps are required to do this:

(i) create a Formal Multiple FSMs Model,
(ii) manually analyze the latter's Reachable State Space, based on permissible FSMs system behavior, and verify the aforementioned key properties and
(iii) derive an PTnet model, of identical behavior, and 1-1 and onto with the FSMs system, and
(iv) lastly, determine the resultant PTnet class, and demonstrate that the net is well-formed (using PIPE2's relevant static verification Algorithms)

All of these steps should be performed with the aid of the PIPE2 tool, *i.e.* to describe the Models, and to simulate with ease. Obviously, PIPE2 will not be able to simulate a set of FSMs properly, but each FSM will be able to be simulated manually as part of the whole, based on the current state(s) of the others.

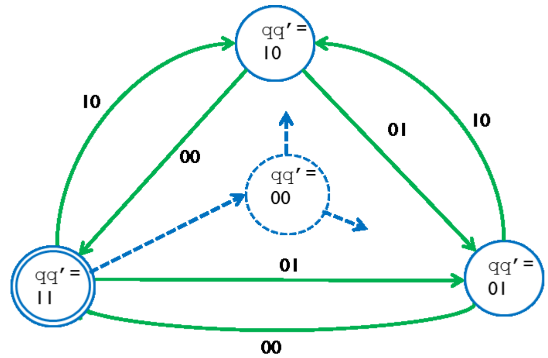


*Figure 2 - Set-Reset Latch Three State FSM Model*

To help you derive the multiple FSMs model, Figure 2 illustrates the Three State FSM model which includes three possible states for the Q, Q' combination, 10 (Q Set), 01 (Q Reset) and 11 (both Q and Q' Set). These are entered, based on the respective input combinations, *i.e.* 10 or 01 respectively. When in state 11, only transitions to 10 and 01 are deterministic, *i.e.* a Set or Reset, and this is illustrated by the fourth, dotted state. Figure 3 shows a possible decomposition of the FF into a set of Three Set-Reset Latches. Note that even though the bottom latch is asymmetric, as the top NAND has three inputs, this only modifies the Set function.

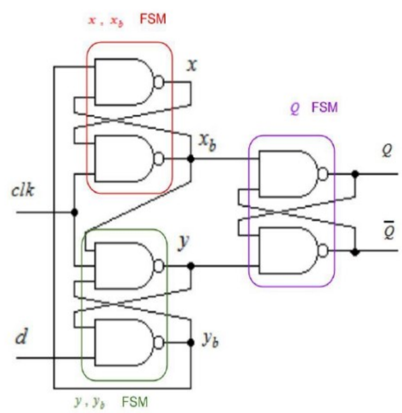


*Figure 3 - Edge-Triggered Flip-Flop Decomposition into a set of Three Set-Reset Latches*

Perform all of the above steps, using PIPE2 to help you as much as possible. Write a short report documenting the analysis, showing each model, how you derived it, how you simulated and analyzed it in PIPE2.

# 2 Assignment 2 – PTnet to Multiple FSMs Conversion

In Assignment 2, the goal is to implement a PTnet using multiple FSMs as a model in the route to implementation. Figure 4 and Figure 5 illustrate PTnet Models for 4-phase to 2-phase and 2-phase to 4-phase protocol converters respectively. Both Models only specify the handshake signals, for simplicity. It is possible to extend them to include data, *i.e.* a data latch storing a bundled-data value, and then passing it on, based on the specific protocol conversion.

The goal of Assignment 2 is to refine the two PTnet Models enough for them to be directly implementable, without state space explosion, *i.e.* reverting to the reachability graph or monolithic FSM state space. This will require a total of six steps:

(i) Create the PTnets in PIPE2
(ii) Perform static verification that the two PTnets are well-formed
(iii) Generate the reachability graph for each of them, and measure the total number of its states
(iv) Extract an S-cover from the PTnets, based on the `get_minimal_siphon()` Algorithm we discussed in the Lectures
(v) Identify the synchronized places, transitions in the S-components of the S-cover
(vi) Convert each S-component into a proper FSM, *i.e.* enrich the transition functions adding inter-FSM synchronization, so that each FSM may be independently implemented
(vii) Perform any obvious vertical (intra-FSM), or horizontal (inter-FSM) state minimizations
(viii) Measure the total number of states of the FSM system and contrast with the state size of the reachability graph

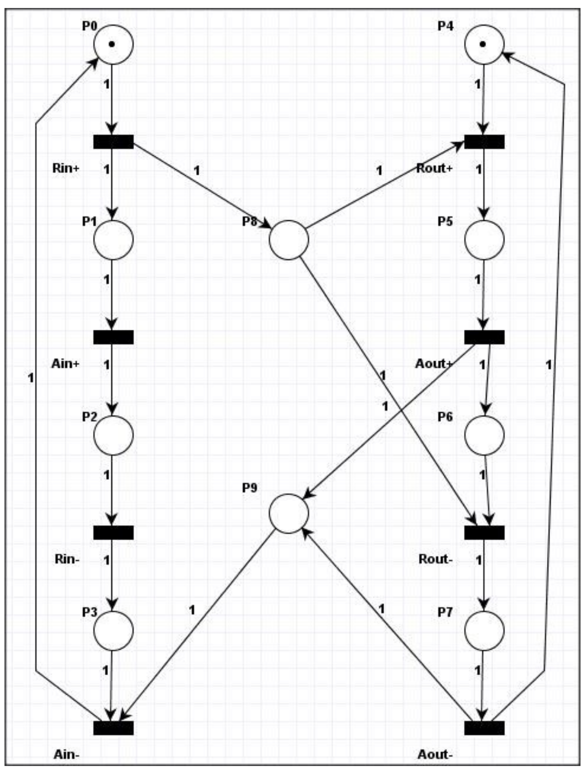


*Figure 4 - 4-Phase to 2-Phase, Handshake only, Protocol Converter*

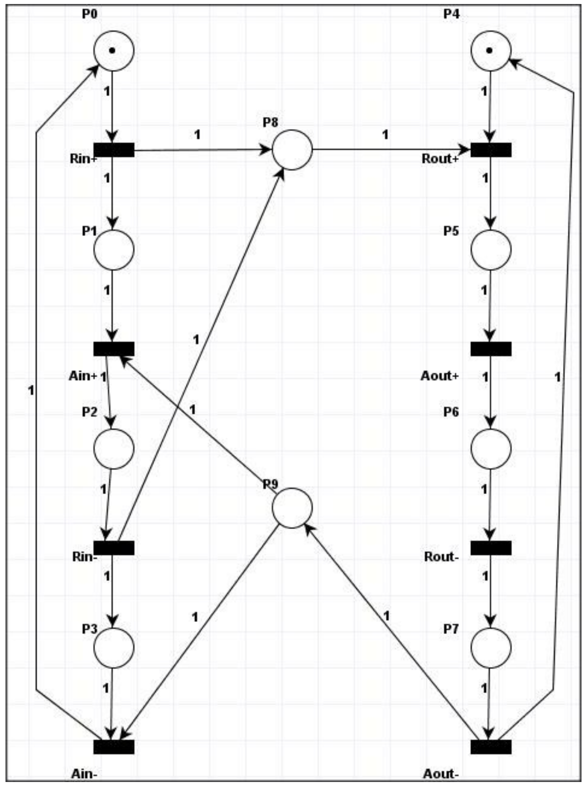


*Figure 5 - 2-Phase to 4-Phase, Handshake only, Protocol Converter*

Write a short report documenting the PTnet to MSFSM flow you followed, *i.e.* all of the above steps, for the two PTnets, showing each step in detail. Again, use PIPE2 as much as possible to help you.