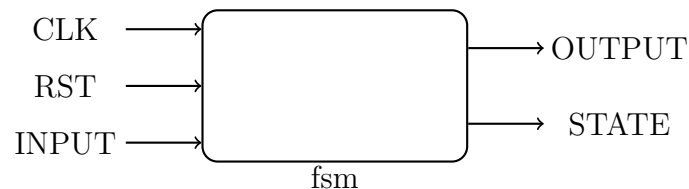


## Mealy FSM (Finite State Machine)

Ihre Aufgabe ist es, das Verhalten einer Entity namens „fsm“ zu programmieren. Die Entity ist in der angehängten Datei „fsm.vhdl“ deklariert und hat folgende Eigenschaften:

- Eingang: CLK vom Typ std\_logic
- Eingang: RST vom Typ std\_logic
- Eingang: INPUT vom Typ std\_logic\_vector mit einer Länge von 2
- Ausgang: OUTPUT vom Typ std\_logic\_vector mit einer Länge von 2
- Ausgang: STATE vom Typ fsm\_state



Verändern Sie die Datei „fsm.vhdl“ nicht!

Die Entity „fsm“ soll das Verhalten einer deterministischen **MEALY** Finite State Machine wiedergeben und dabei folgende Eigenschaften erfüllen:

- Ein Verhalten entsprechend dem Zustandsübergangsdiagramm in Abbildung 1. Der Eingang, welcher zu dem nächsten Zustand führt, und der Ausgang im nächsten Zustand sind an den Kanten angegeben.
- Übergänge mit steigender Flanke des Taktsignals (Port CLK; dies ist ein rechteckiges Taktsignal)
- Synchrones Design: Alle neuen Ausgänge (Port STATE und Port OUTPUT) müssen mit der steigenden Taktflanke gesetzt werden.
- Der aktuelle Zustand muss am Ausgangsport STATE ausgegeben werden.
- Bei einem synchronen Reset (Port RST=1, steigende Flanke des Taktsignals CLK) soll die State Machine auf den initialen Zustand (Anfangszustand) gesetzt werden: STATE = START, OUTPUT = "00".
- Für Eingänge, für die kein Übergang im Diagramm definiert ist, soll folgendes Verhalten implementiert werden: Zustand soll sich nicht ändern, OUTPUT = "00".

Programmieren Sie dieses Verhalten in der angehängten Datei „fsm\_beh.vhdl“.

Der Typ fsm\_state ist in der angehängten Datei „fsm\_package.vhdl“ deklariert. Die dafür benötigten Pakete sind in den Dateien „fsm\_beh.vhdl“ und „fsm.vhdl“ bereits importiert.

Um Ihre Lösung abzugeben, senden Sie ein E-Mail mit dem Betreff „Result Task 4“ und Ihrer Datei „fsm\_beh.vhdl“ an [vhdl-mc+e384@tuwien.ac.at](mailto:vhdl-mc+e384@tuwien.ac.at).

Viel Erfolg und möge die Macht mit Ihnen sein.

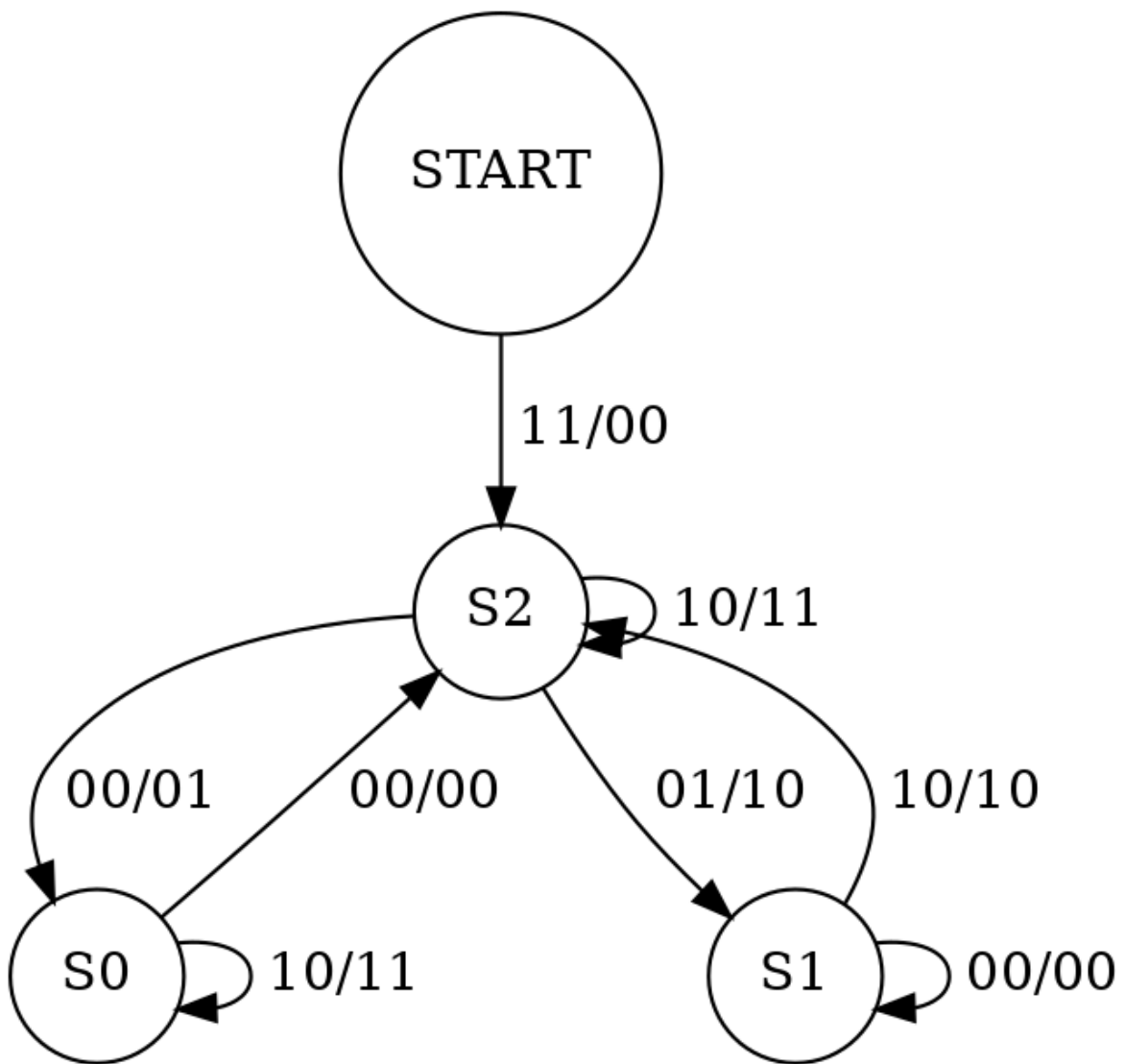


Abbildung 1: Zustandsübergangsdiagramm, die Kanten sind mit *Eingang / nächster Ausgang* beschriftet