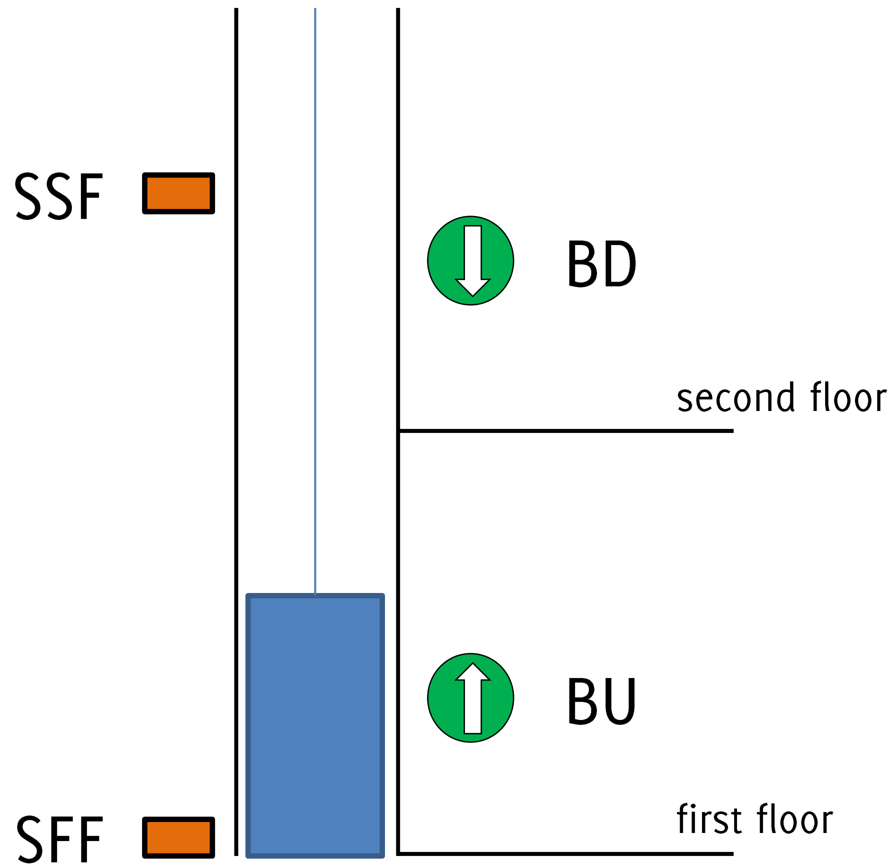
MIDTERM REVIEW:
FINITE STATE MODELING

ELEVATOR EXAMPLE (GREATLY SIMPLIFIED)

An elevator operates between the first and second floor of a building. There are no buttons inside the elevator, only a button on the first floor, and a button on the second floor. Sensors detect when the elevator has reached the first or second floor.



DESIRED OPERATION

- If up button is pressed (BTN_UP=1) and elevator is on the 1st floor, activate UP signal (UP=1) to raise elevator.
- If down button is pressed (BTN_DN=1) and elevator is on the 2nd floor, activate DN signal (DN=1) to lower elevator.
- Deactivate the UP signal when the elevator reaches the second floor, as indicated by the second floor sensor (SSF=1)
- Deactivate the DN signal when the elevator reaches the first floor, as indicated by the first floor sensor (SFF=1)
- Ignore button inputs while elevator is in motion.
- Assume sensor on departing floor turns off as soon as elevator motion begins.
- Activate an error signal (ERR=1) and stop the elevator if an unexpected input condition is detected.

INPUTS AND OUTPUTS

Inputs

BU = 1: 1^{st} floor up button pressed; take elevator to 2^{nd} floor

BU = 0: 1^{st} floor up button not pressed

BD = 1: 2^{nd} floor down button pressed; take elevator to 1^{st} floor

BD = 0: 2^{nd} floor down button not pressed

SFF = 1: Elevator on 1^{st} floor

SFF = 0: Elevator not on 1^{st} floor

SSF = 1: Elevator on 2^{nd} floor

SSF = 0: Elevator not on 2^{nd} floor

Outputs

UP = 1: UP signal active; Raise elevator to 2^{nd} floor

UP = 0: UP signal not active

DN = 1: DN signal active; Lower elevator to 1^{st} floor

DN = 0: DN signal not active

ERR = 1: Error condition detected; signal for assistance

ERR = 0: Error condition not detected

STATES?

STATES?

S0: elevator stopped on first floor (FF)

S1: move elevator upward to first floor (UP)

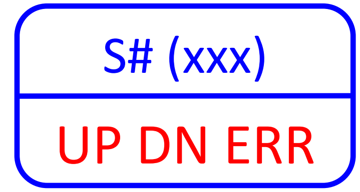
S2: elevator stopped on second floor (SF)

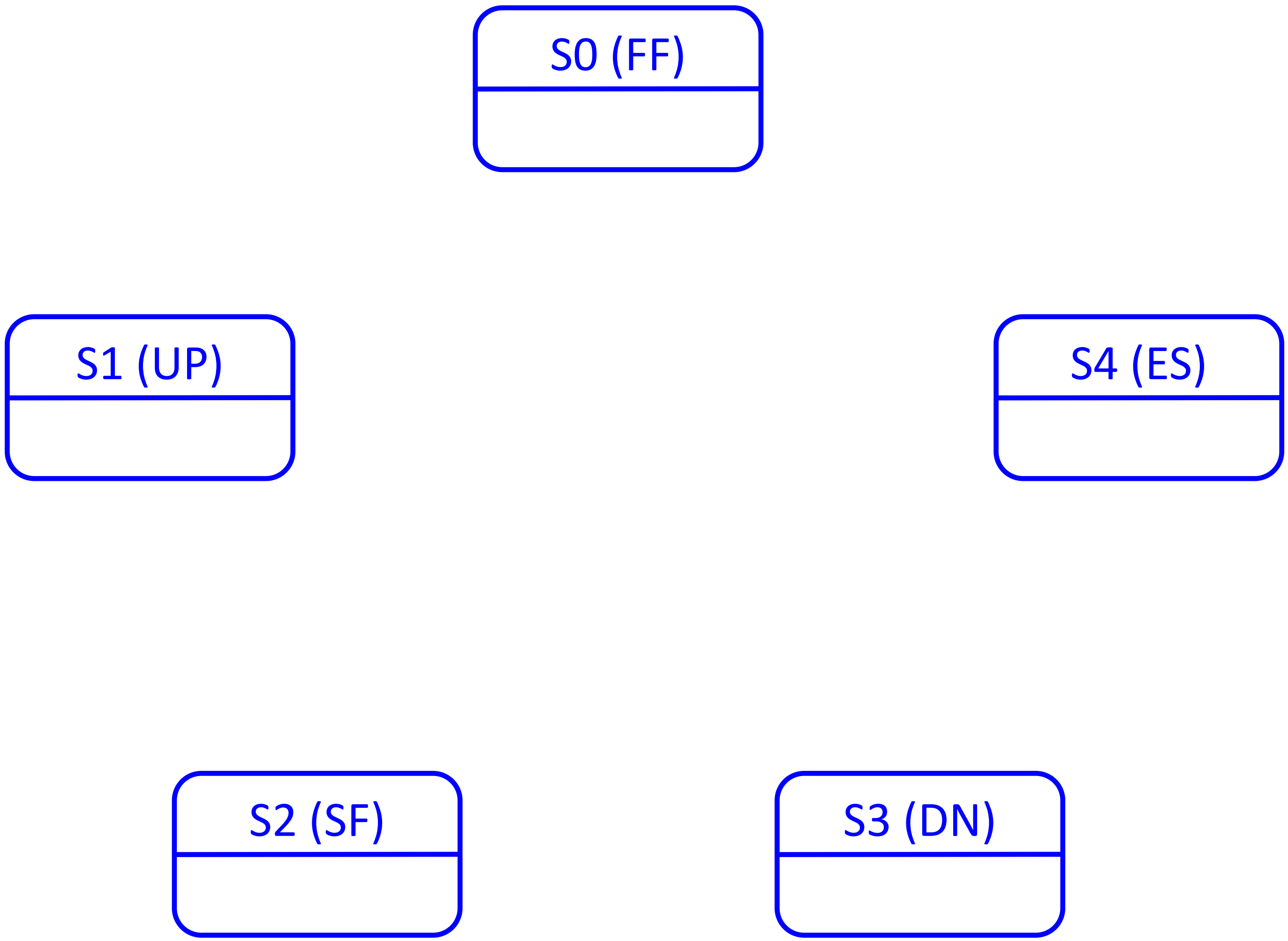
S3: move elevator downward second floor (DN)

S4: emergency shutdown on error (ES)

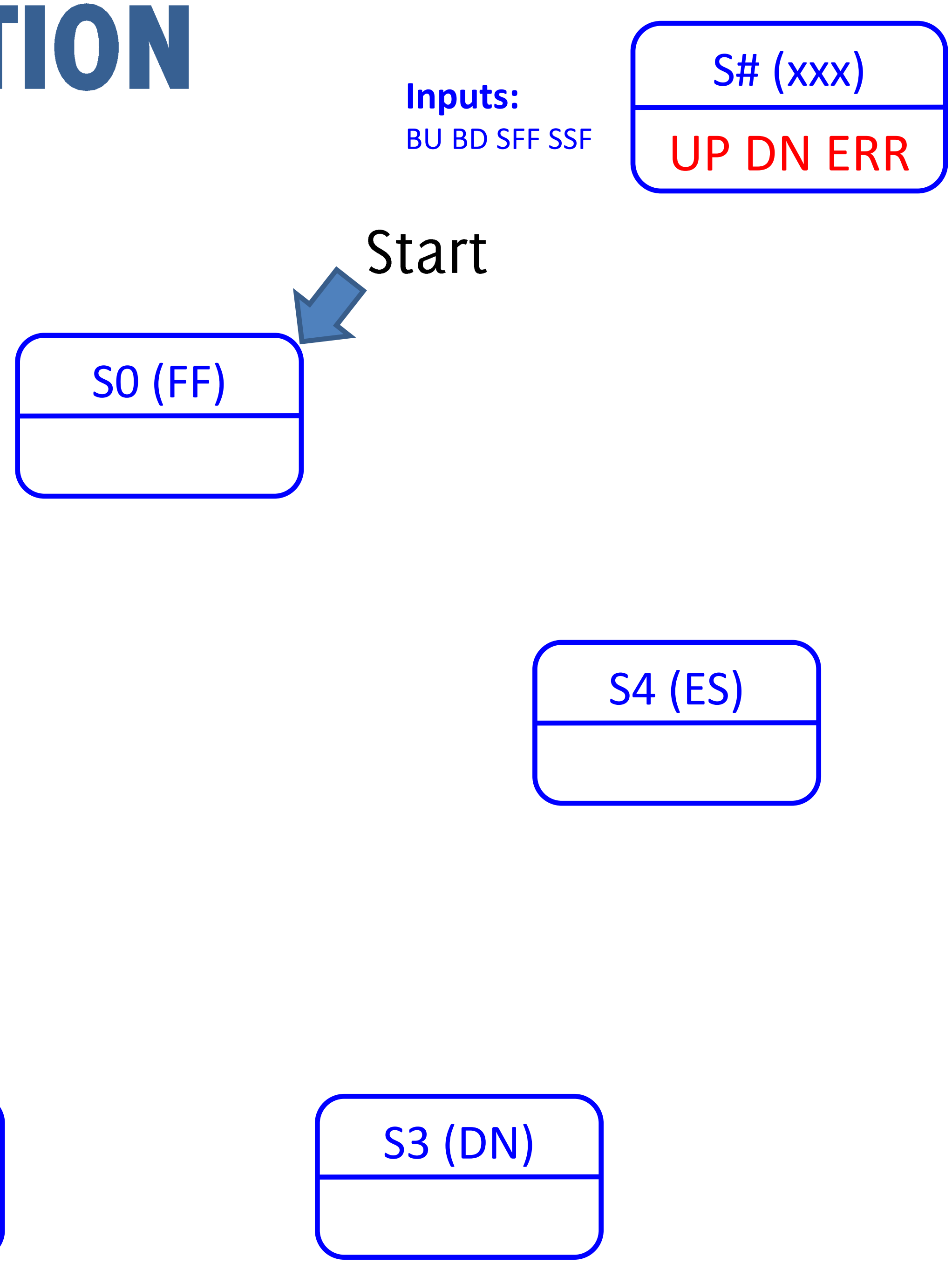
STATE TRANSITION DIAGRAM

Inputs:
BU BD SFF SSF



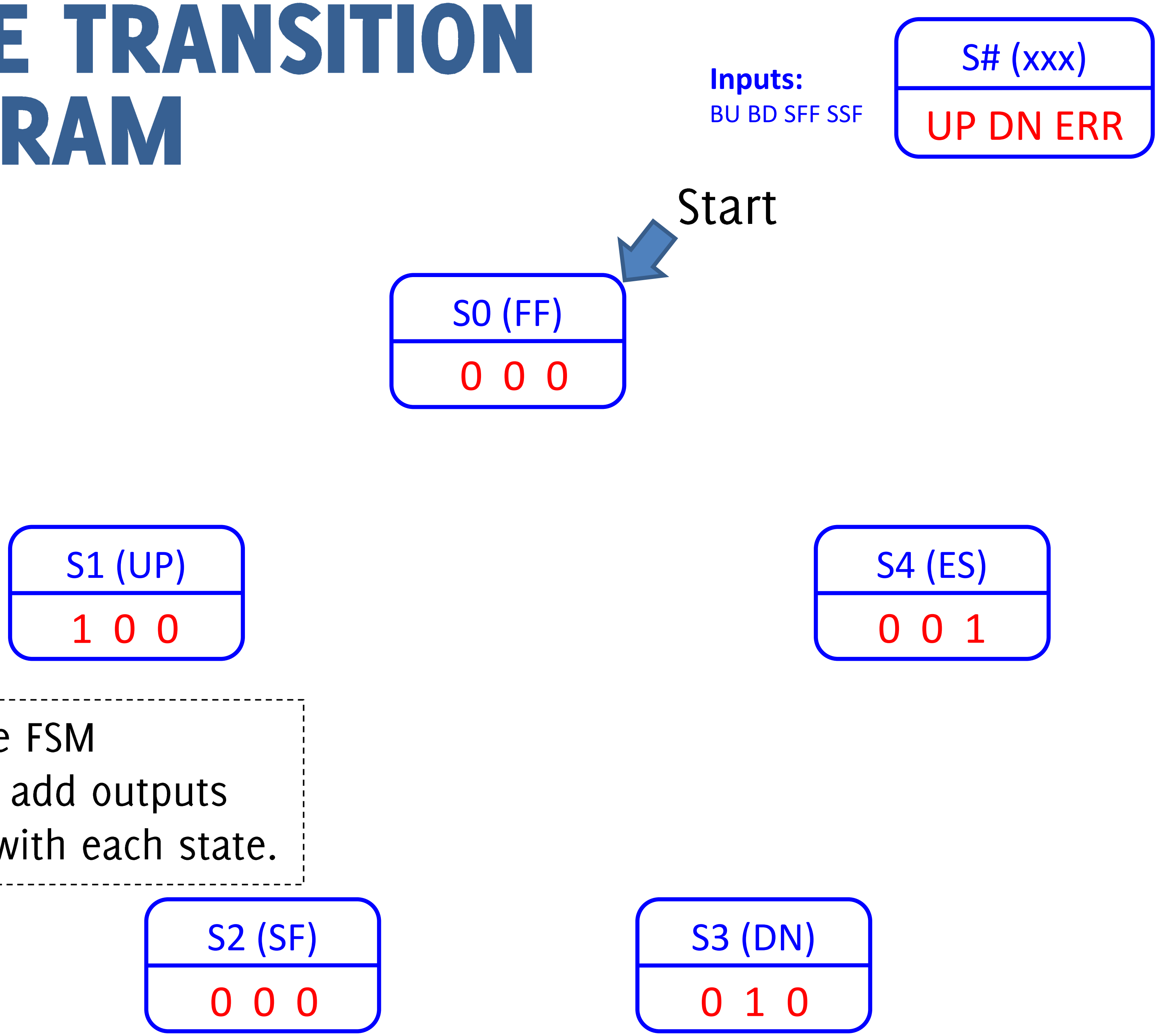


STATE TRANSITION DIAGRAM



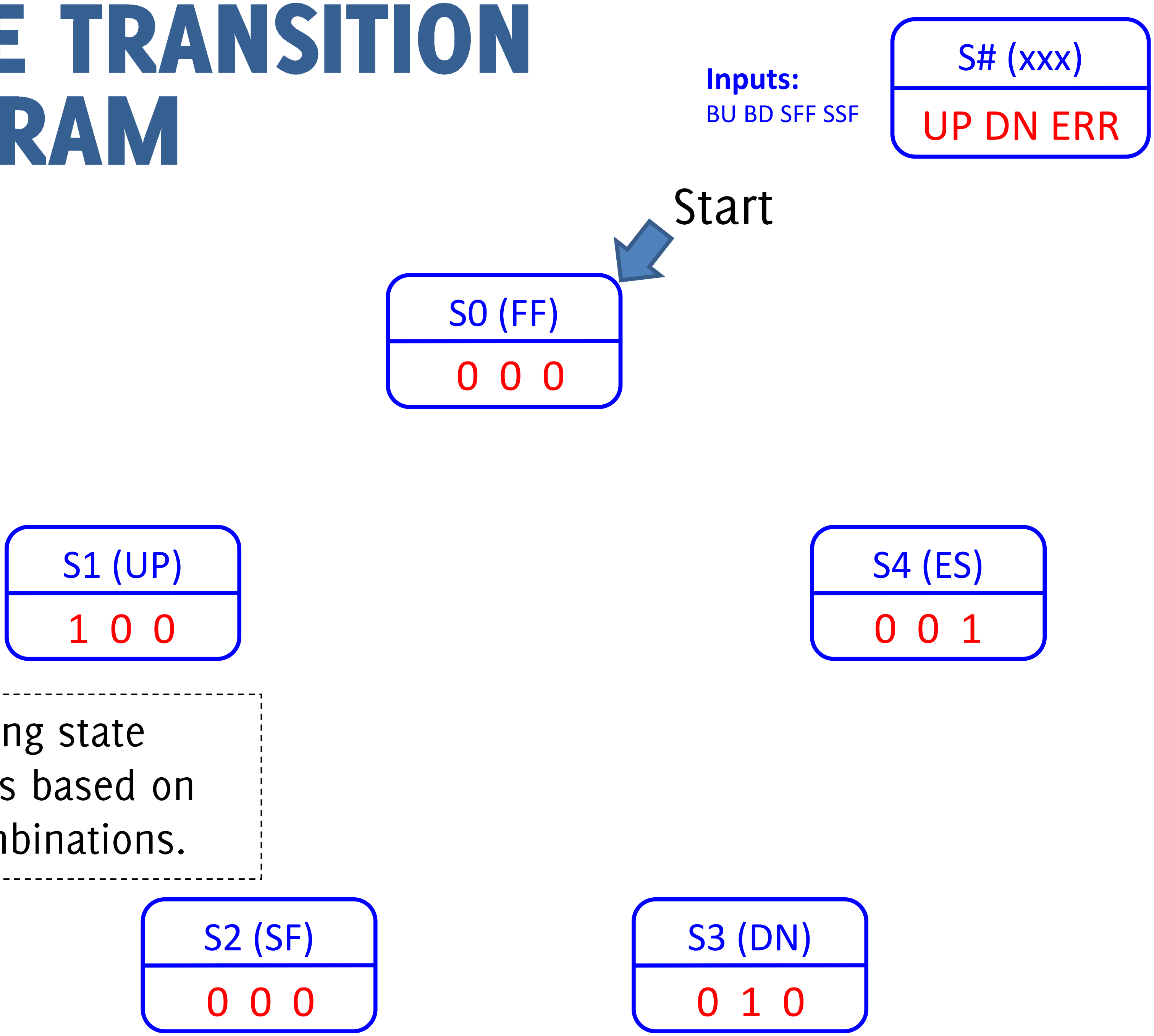
To avoid any confusion, identify the starting state!

STATE TRANSITION DIAGRAM



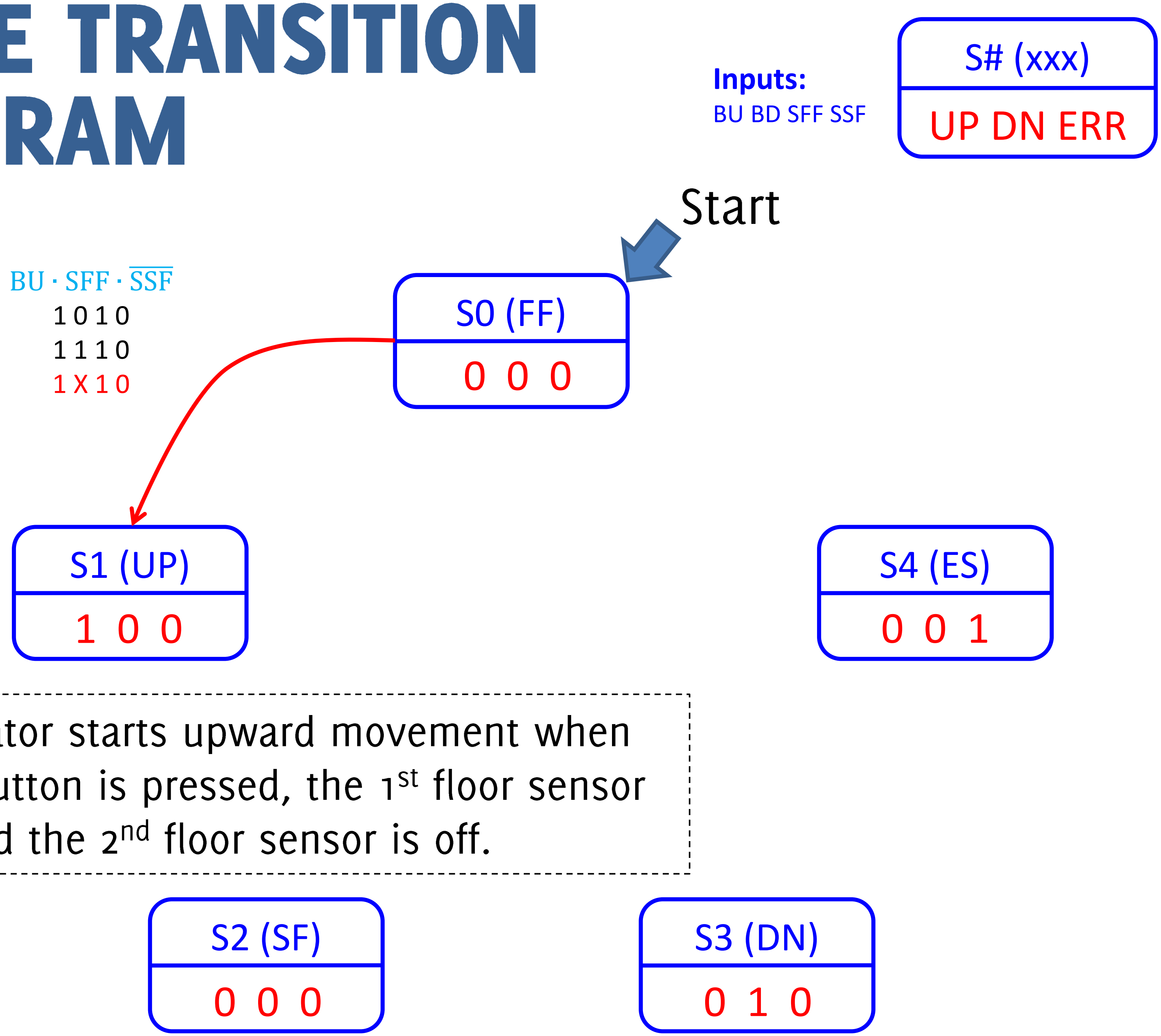
Using Moore FSM convention, add outputs associated with each state.

STATE TRANSITION DIAGRAM



Start adding state transitions based on input combinations.

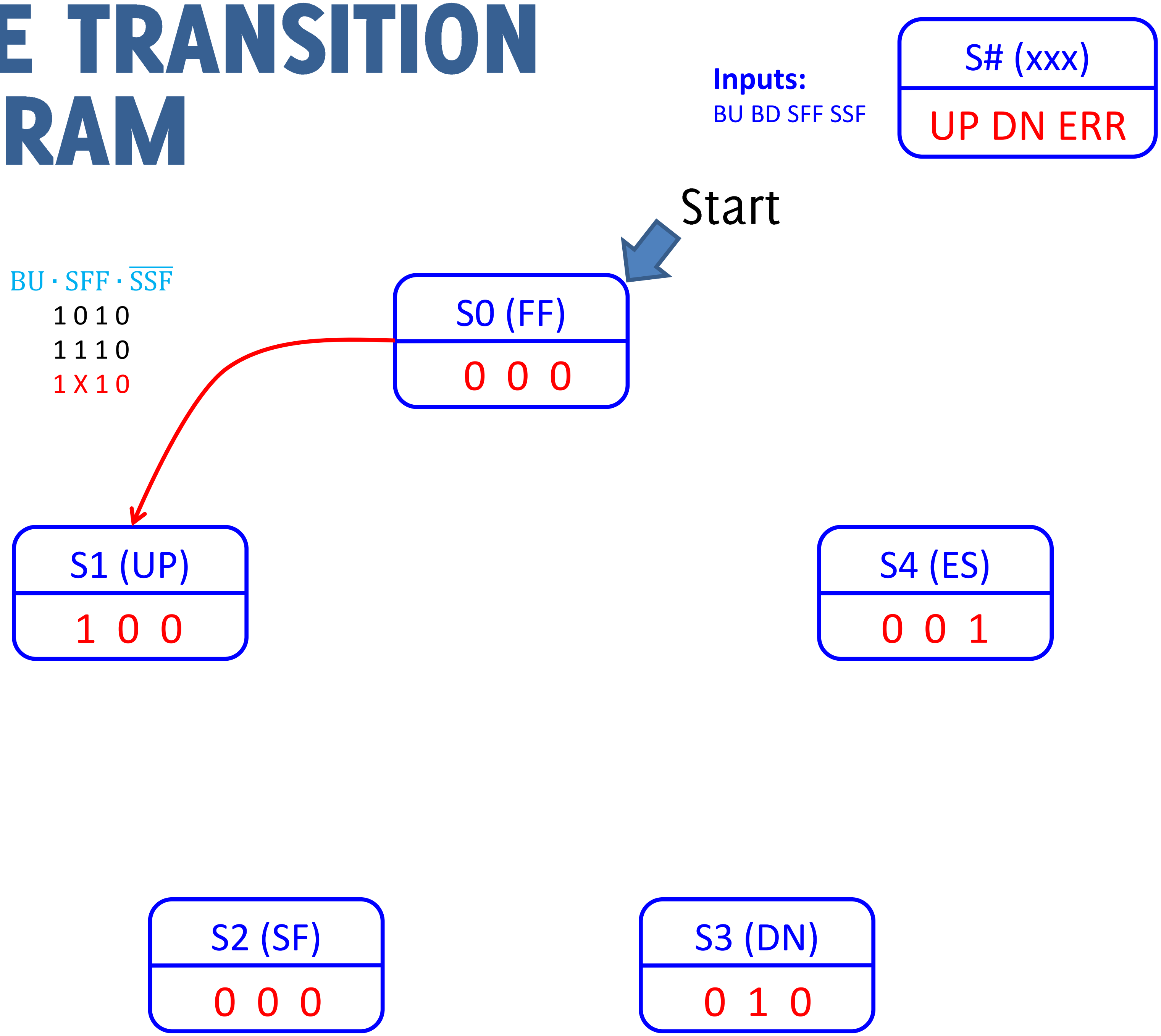
STATE TRANSITION DIAGRAM



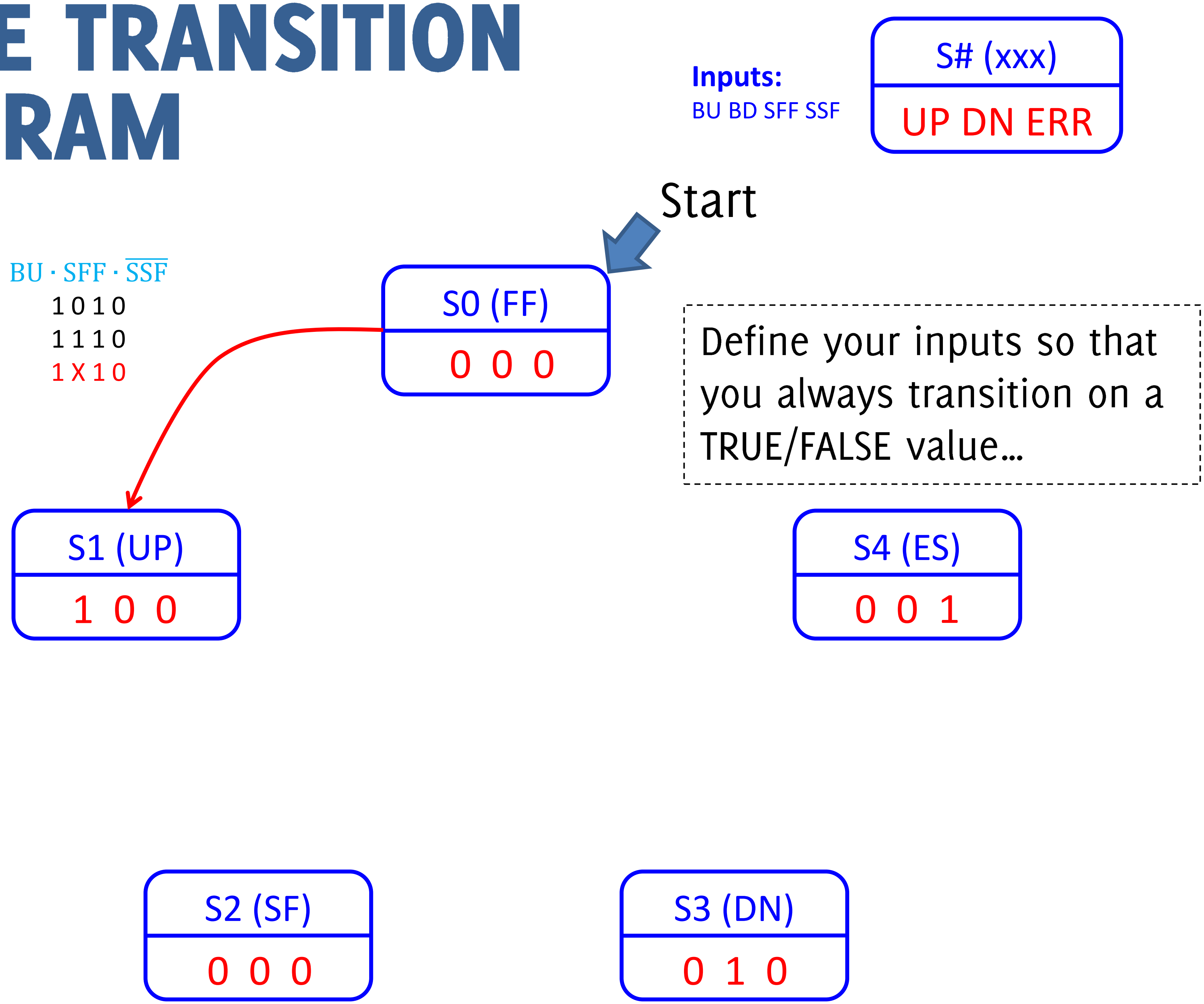
The elevator starts upward movement when the up button is pressed, the 1st floor sensor is on, and the 2nd floor sensor is off.

STATE TRANSITION DIAGRAM

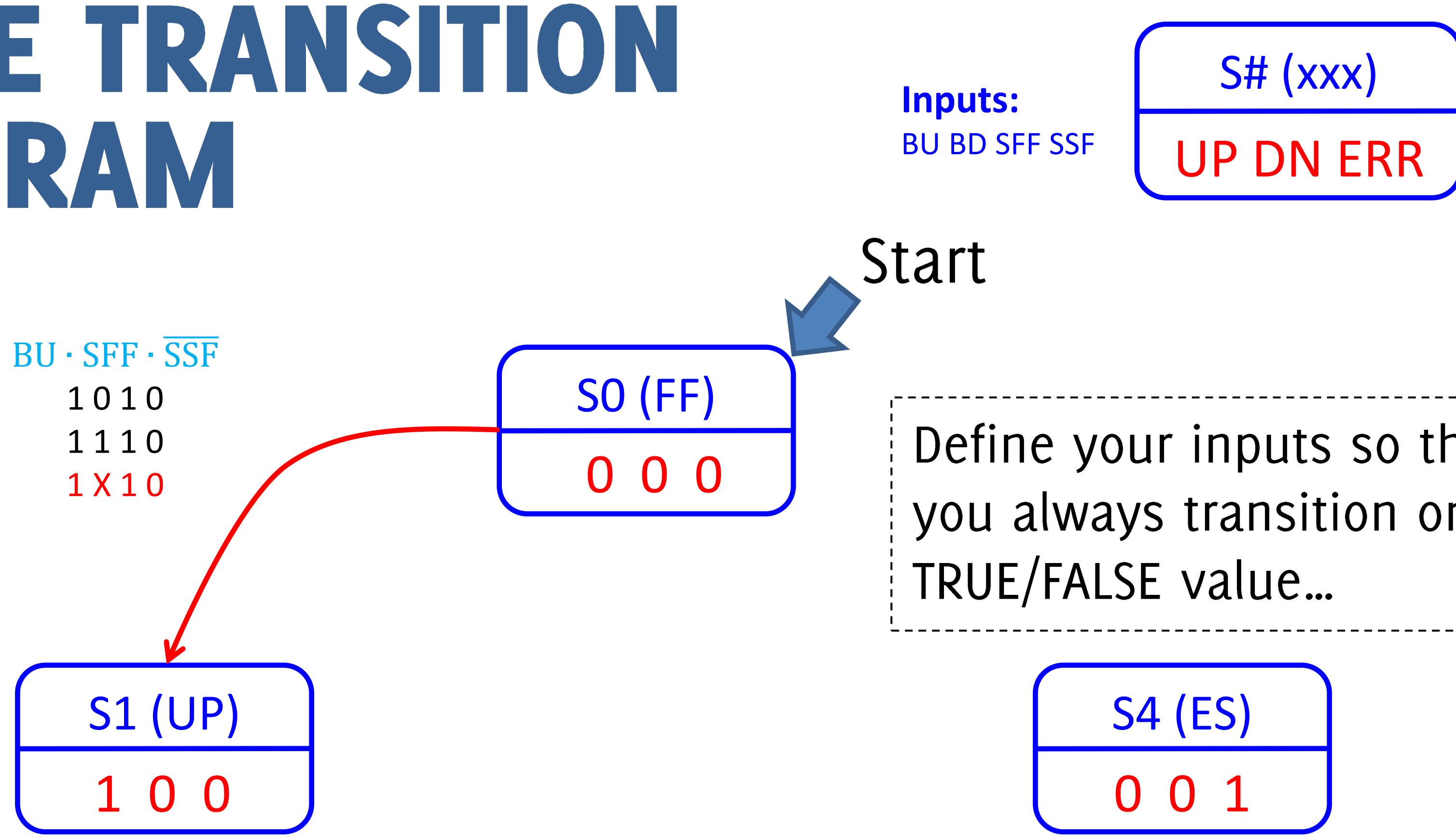
Several different ways of → notating transition inputs



STATE TRANSITION DIAGRAM



STATE TRANSITION DIAGRAM

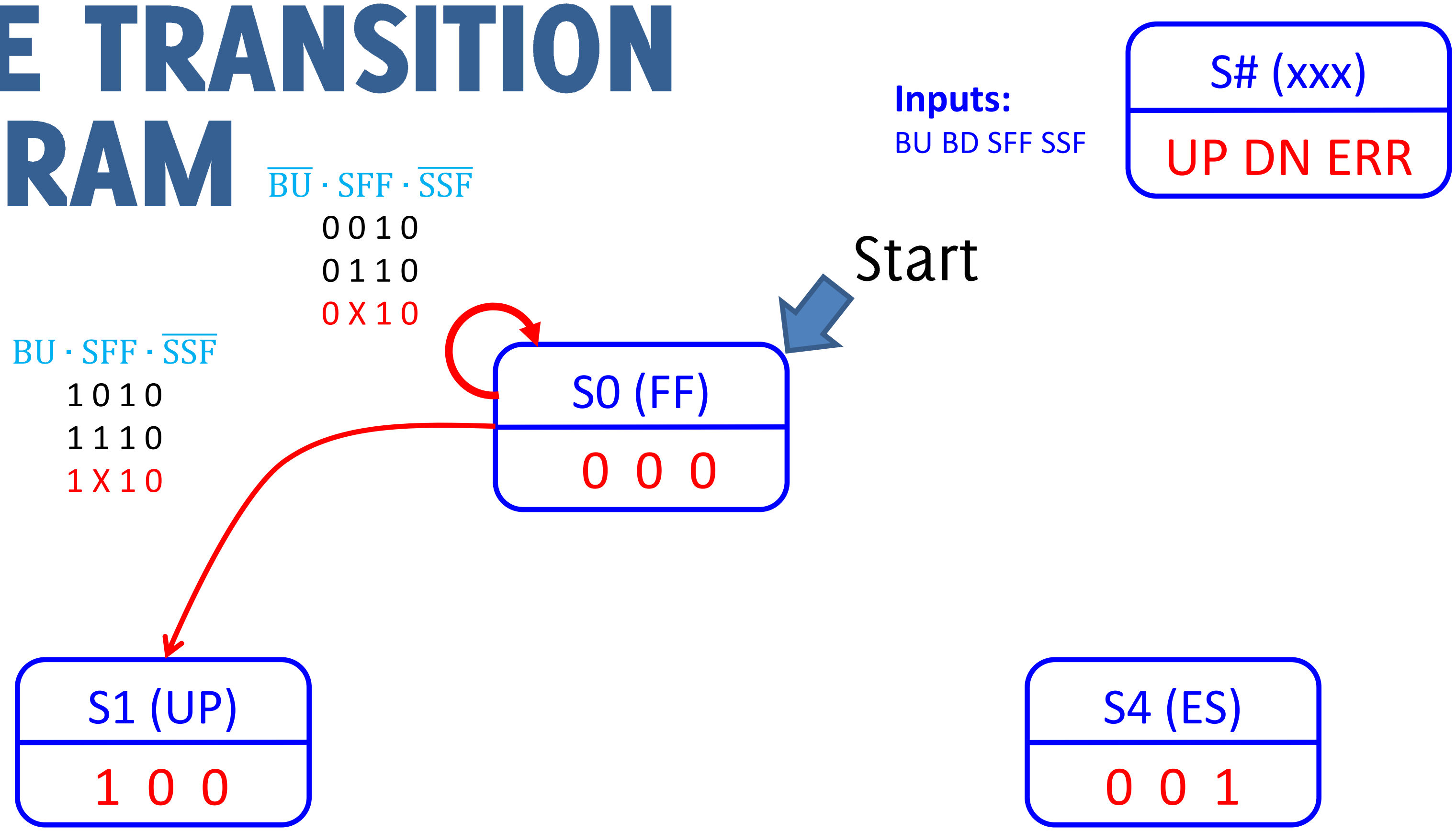


Define your inputs so that you always transition on a TRUE/FALSE value…

For instance, if you have a voltage signal that ranges from 0 to 5V, and you want to transition when the voltage is greater than 3V, then define your input as something like:

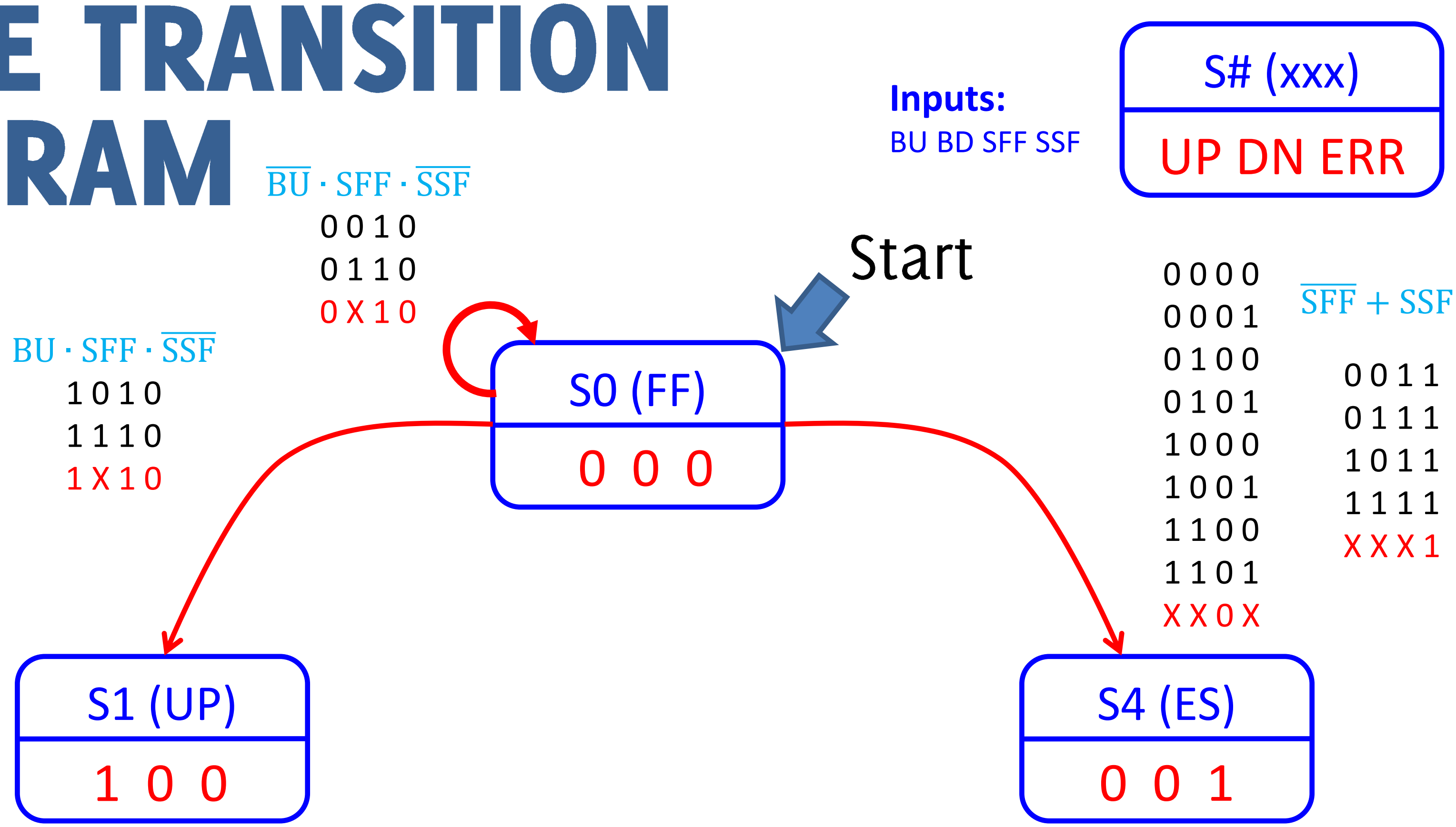
VS = 1; voltage greater than 3V, take action 1
VS = 0; voltage less than or equal to 3V, take action 2

STATE TRANSITION DIAGRAM



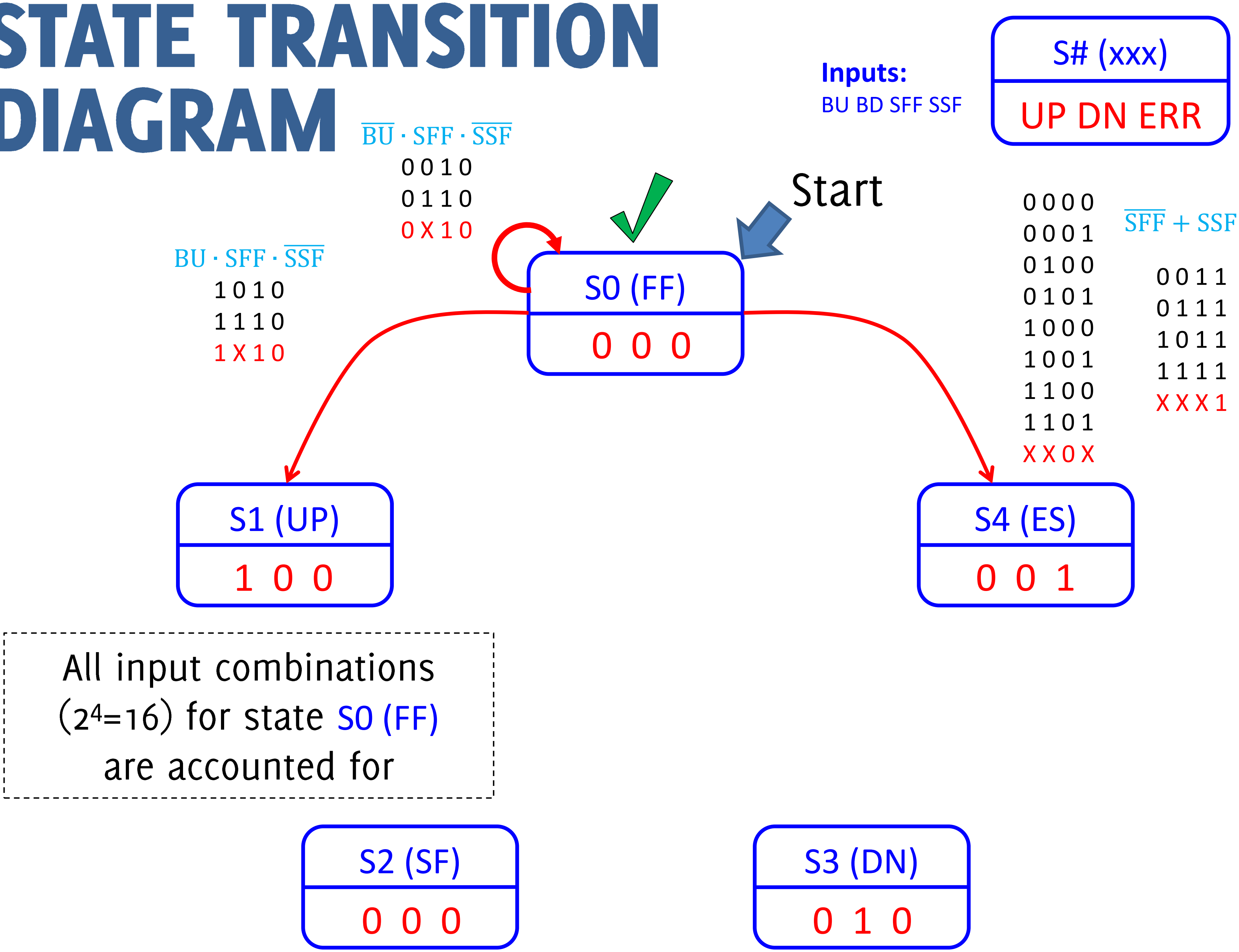
We stay on the ground floor as long as the up button is not pressed, the 1st floor sensor is on, and the 2nd floor sensor is off.

STATE TRANSITION DIAGRAM



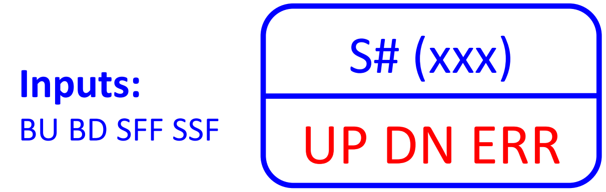
If we are on the ground floor, we should never see the 1st floor sensor be off, or the 2nd floor sensor be active.

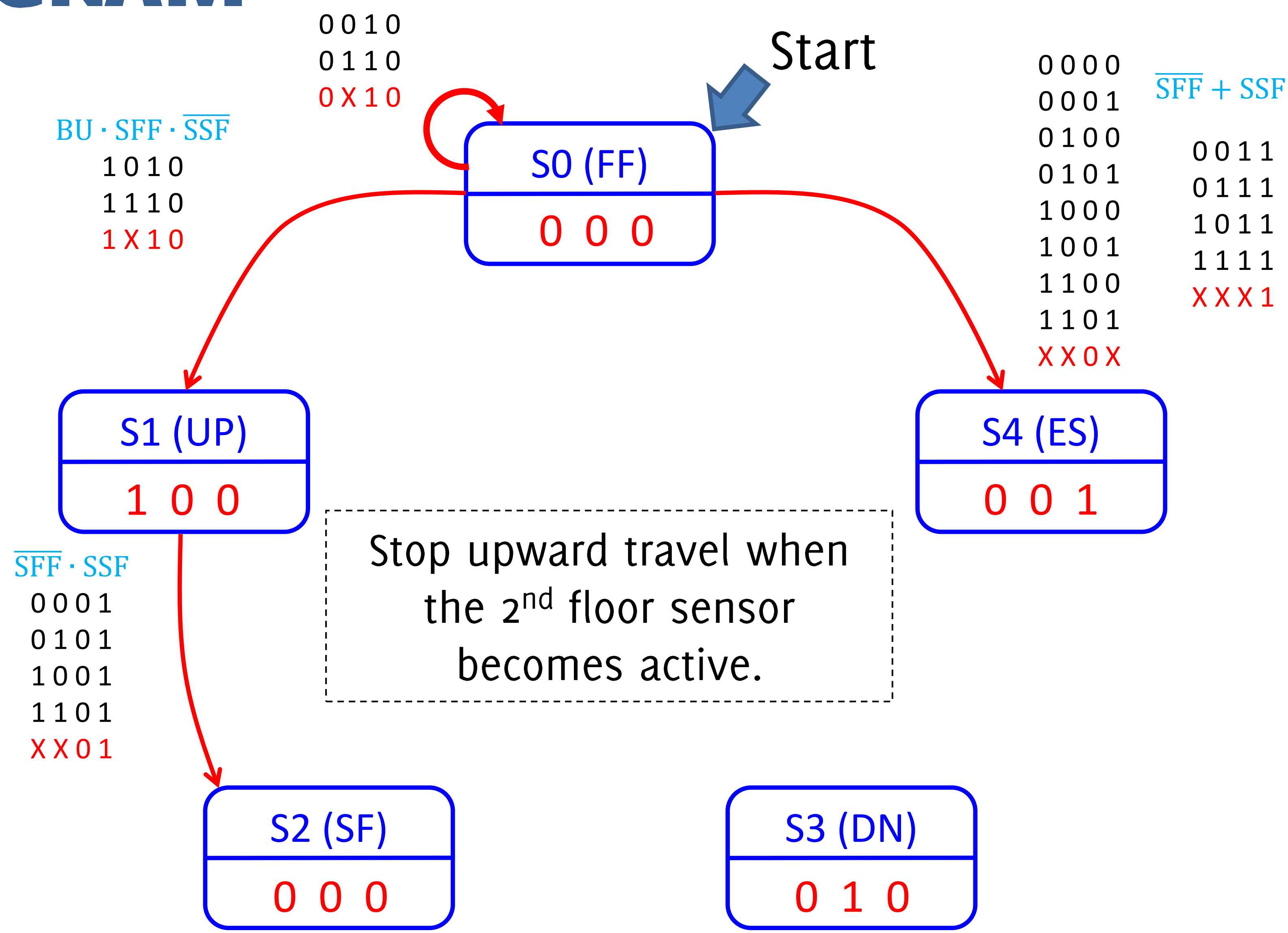
STATE TRANSITION DIAGRAM



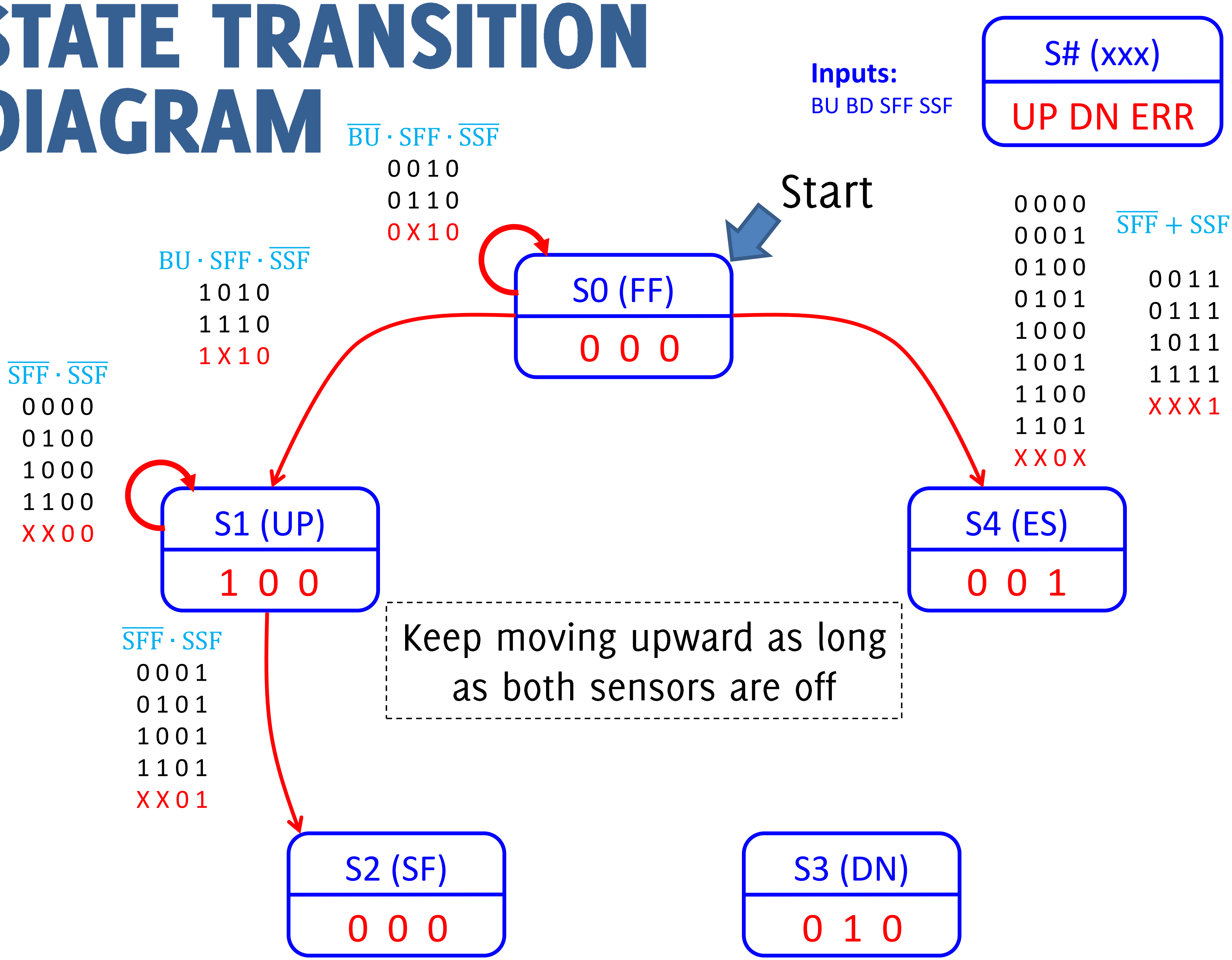
All input combinations (2^4=16) for state S0 (FF) are accounted for

STATE TRANSITION DIAGRAM

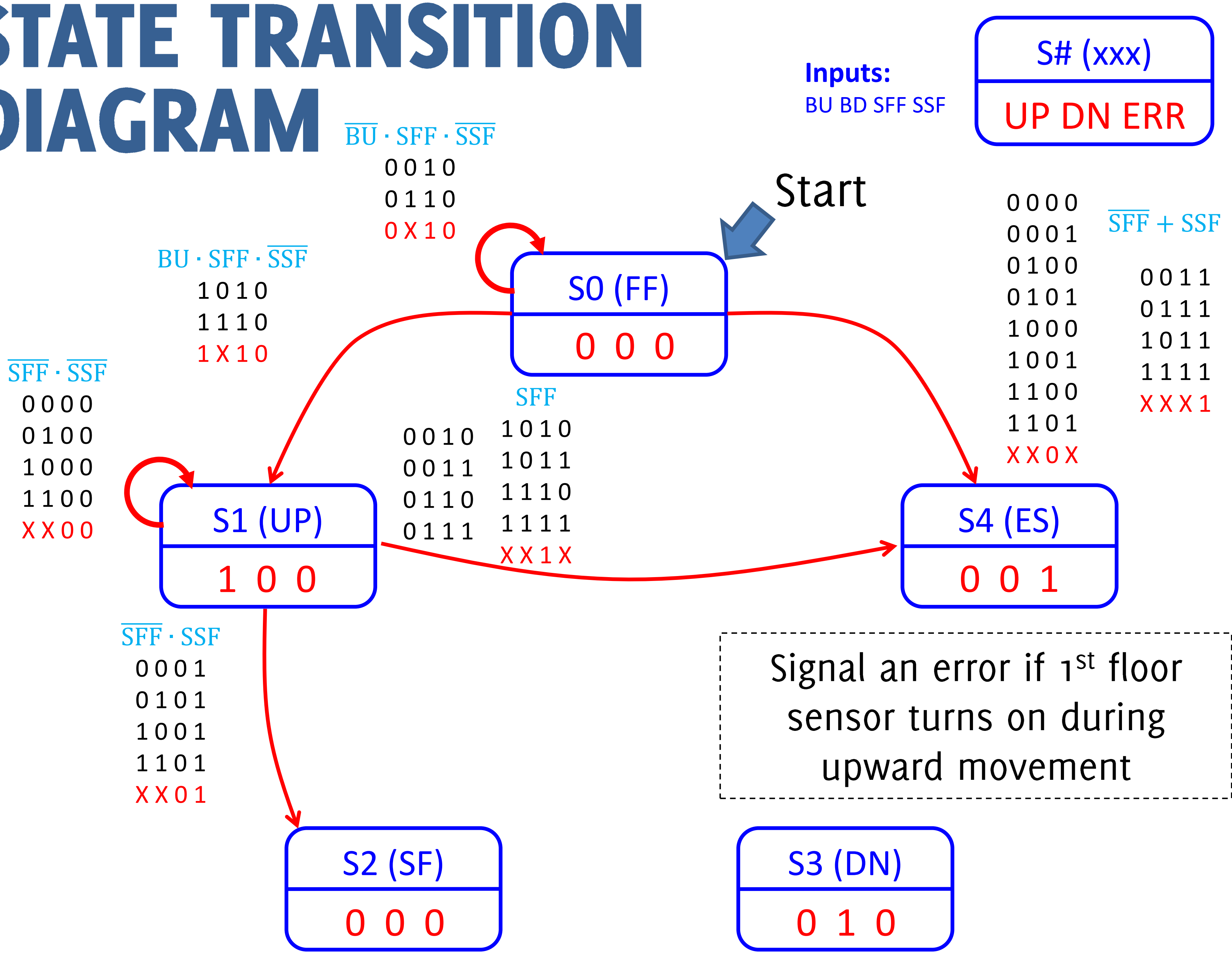




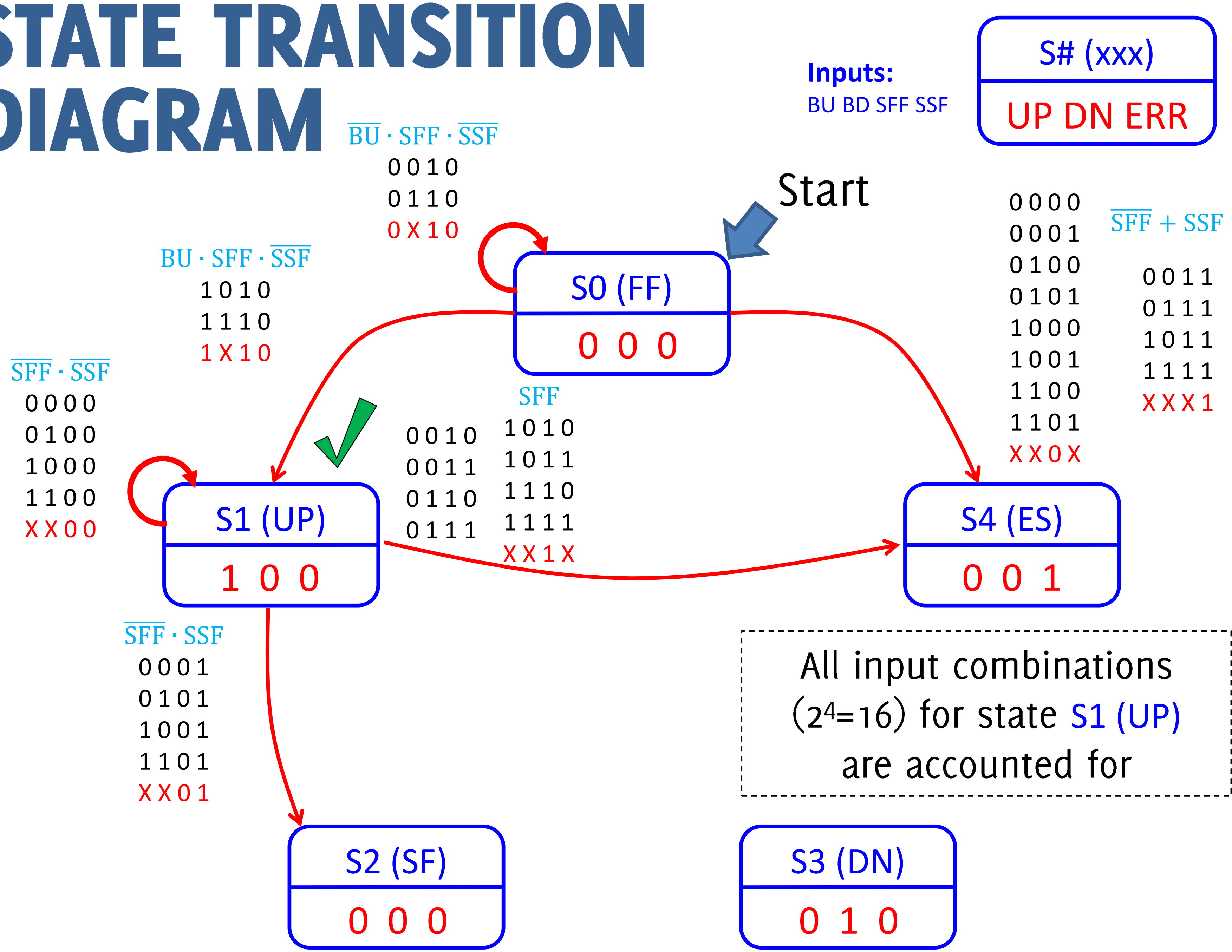
STATE TRANSITION DIAGRAM



STATE TRANSITION DIAGRAM

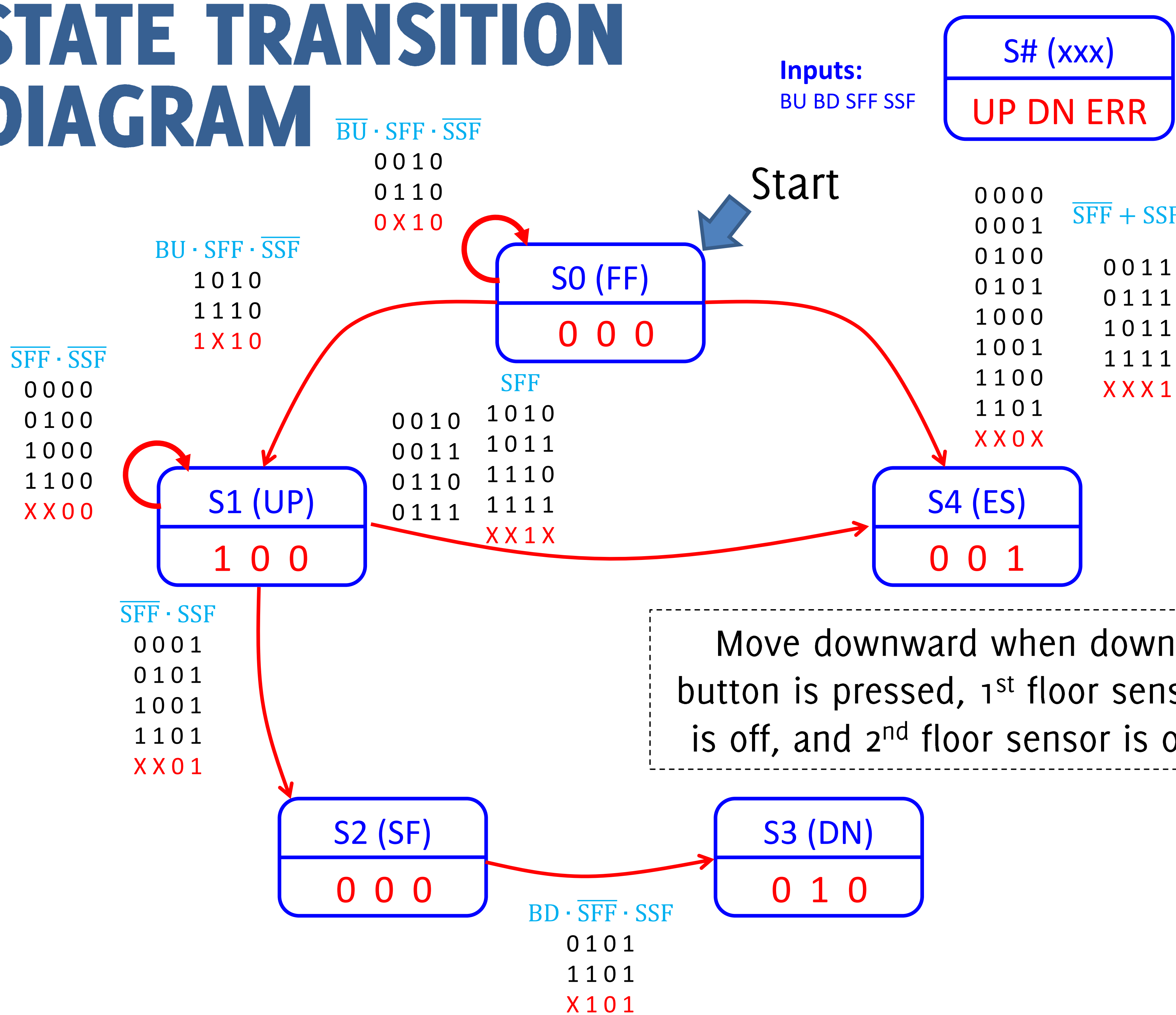


STATE TRANSITION DIAGRAM



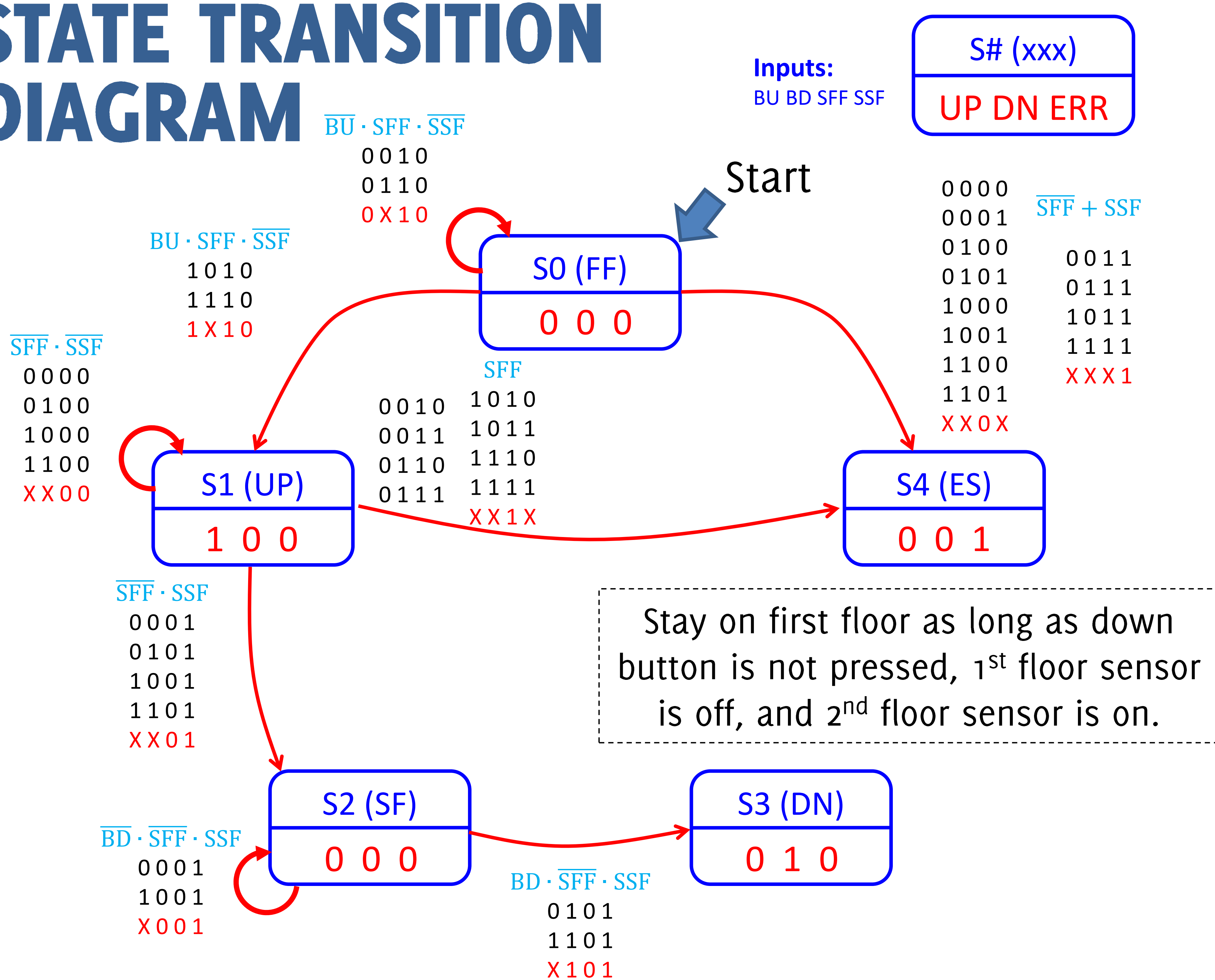
All input combinations (2^4=16) for state S1 (UP) are accounted for

STATE TRANSITION DIAGRAM

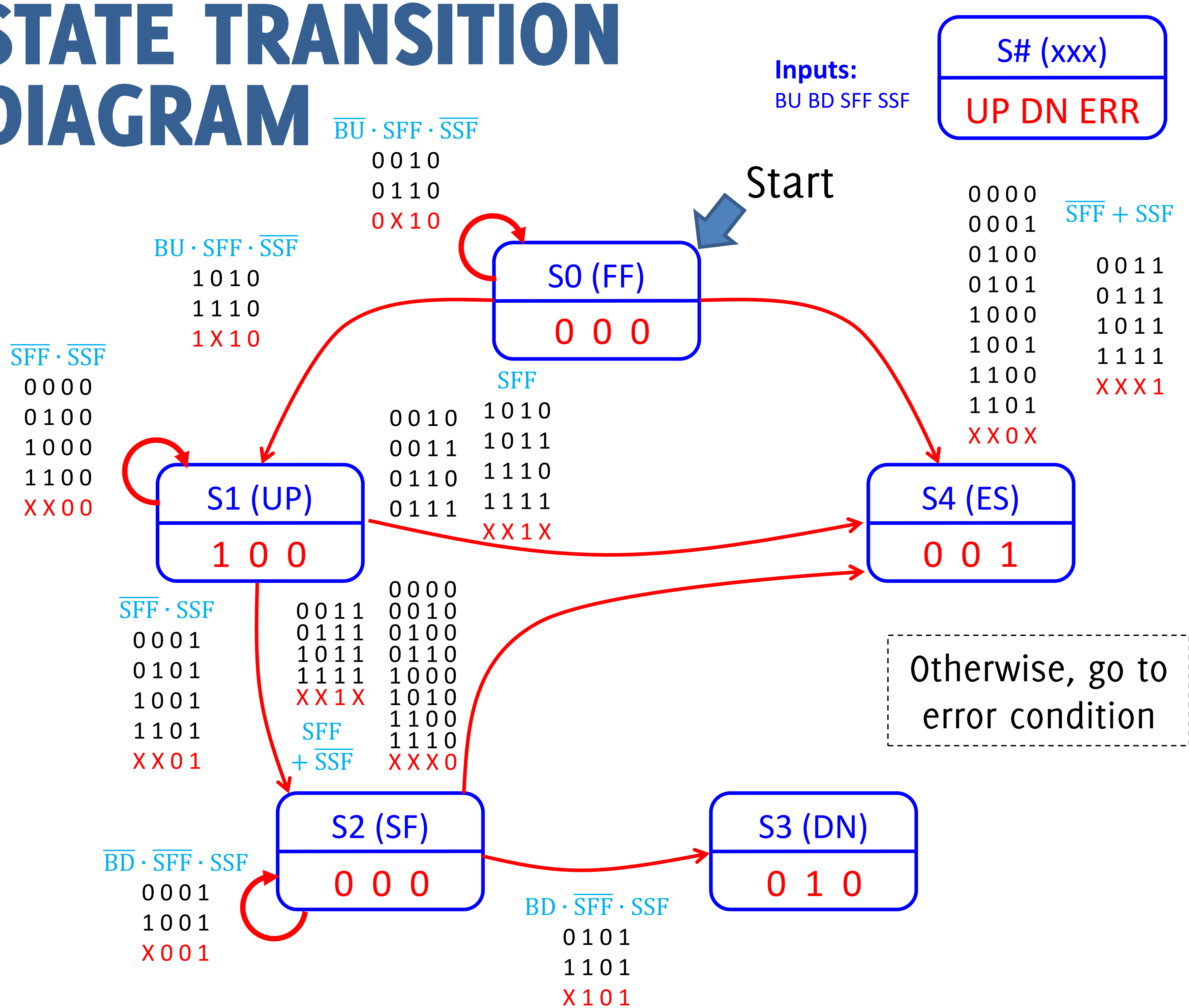


Move downward when down button is pressed, 1st floor sensor is off, and 2nd floor sensor is on

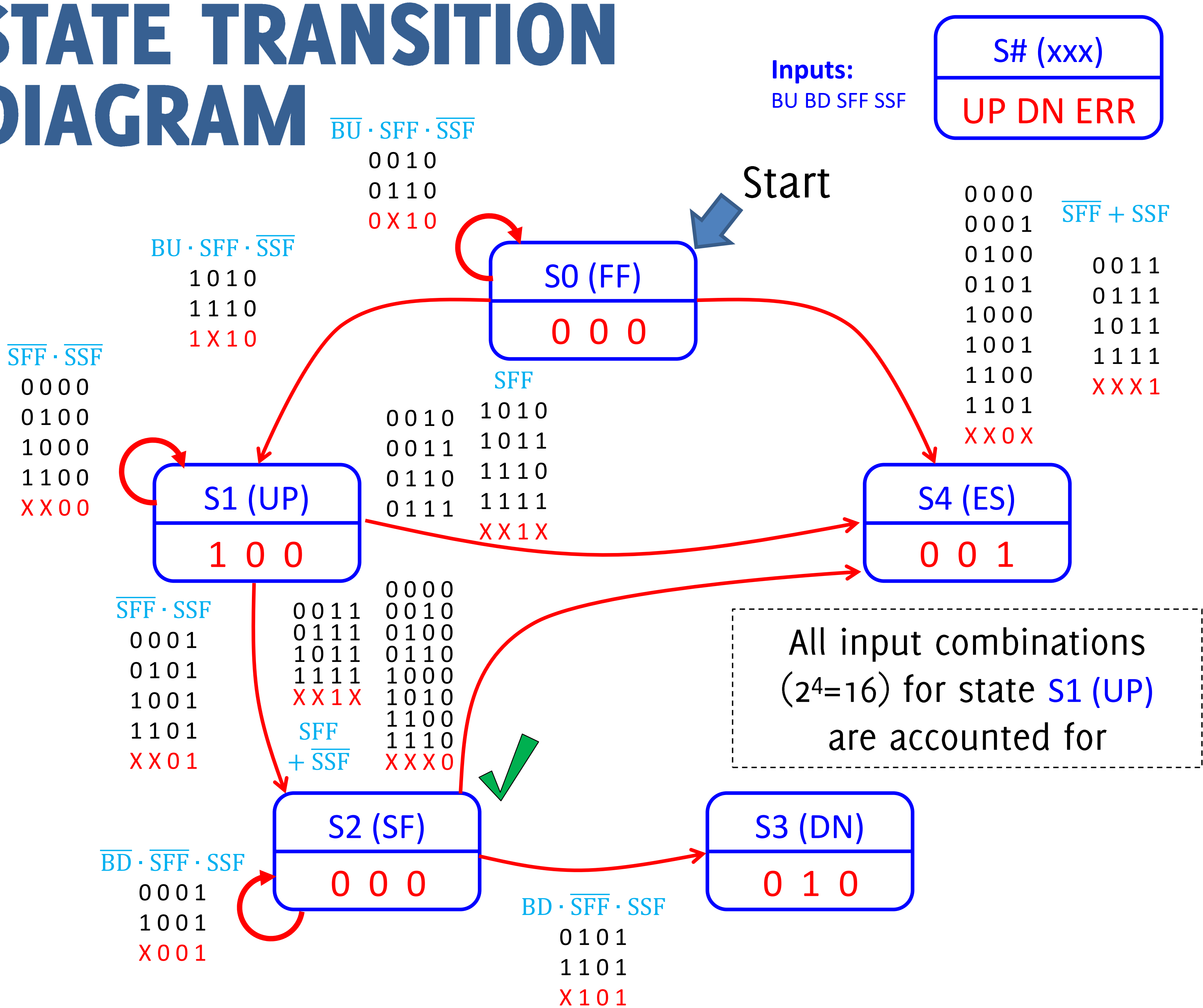
STATE TRANSITION DIAGRAM



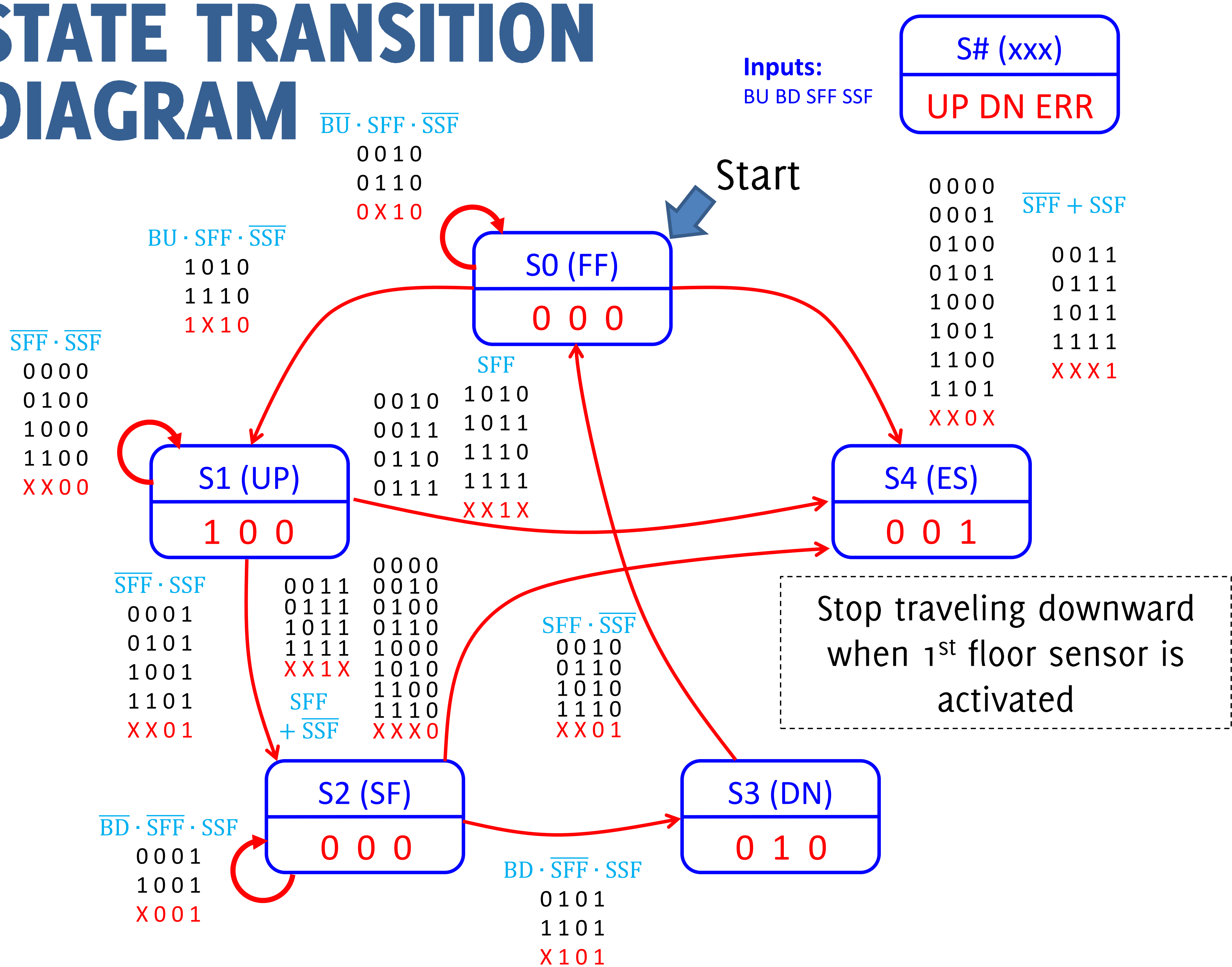
STATE TRANSITION DIAGRAM



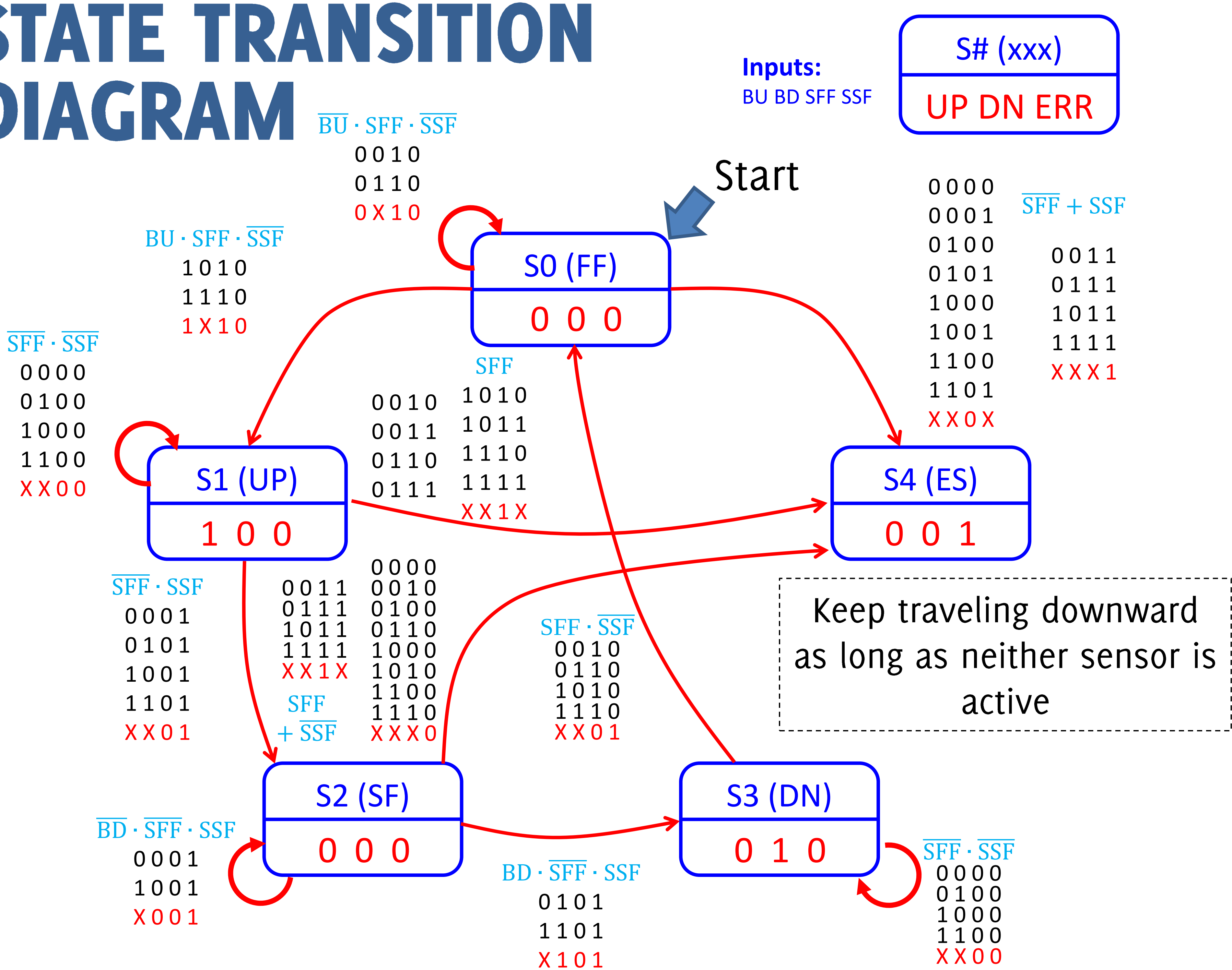
STATE TRANSITION DIAGRAM



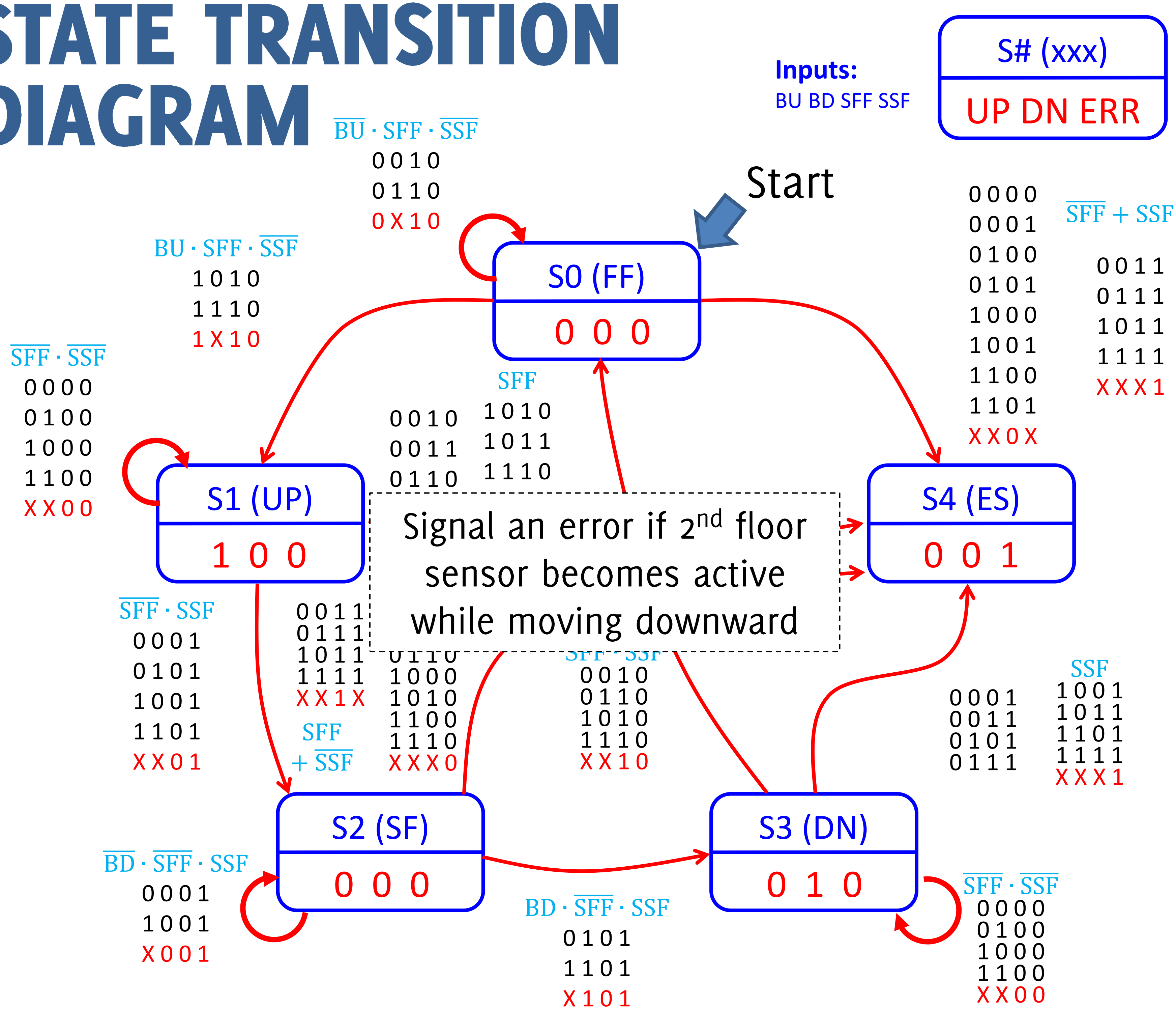
STATE TRANSITION DIAGRAM



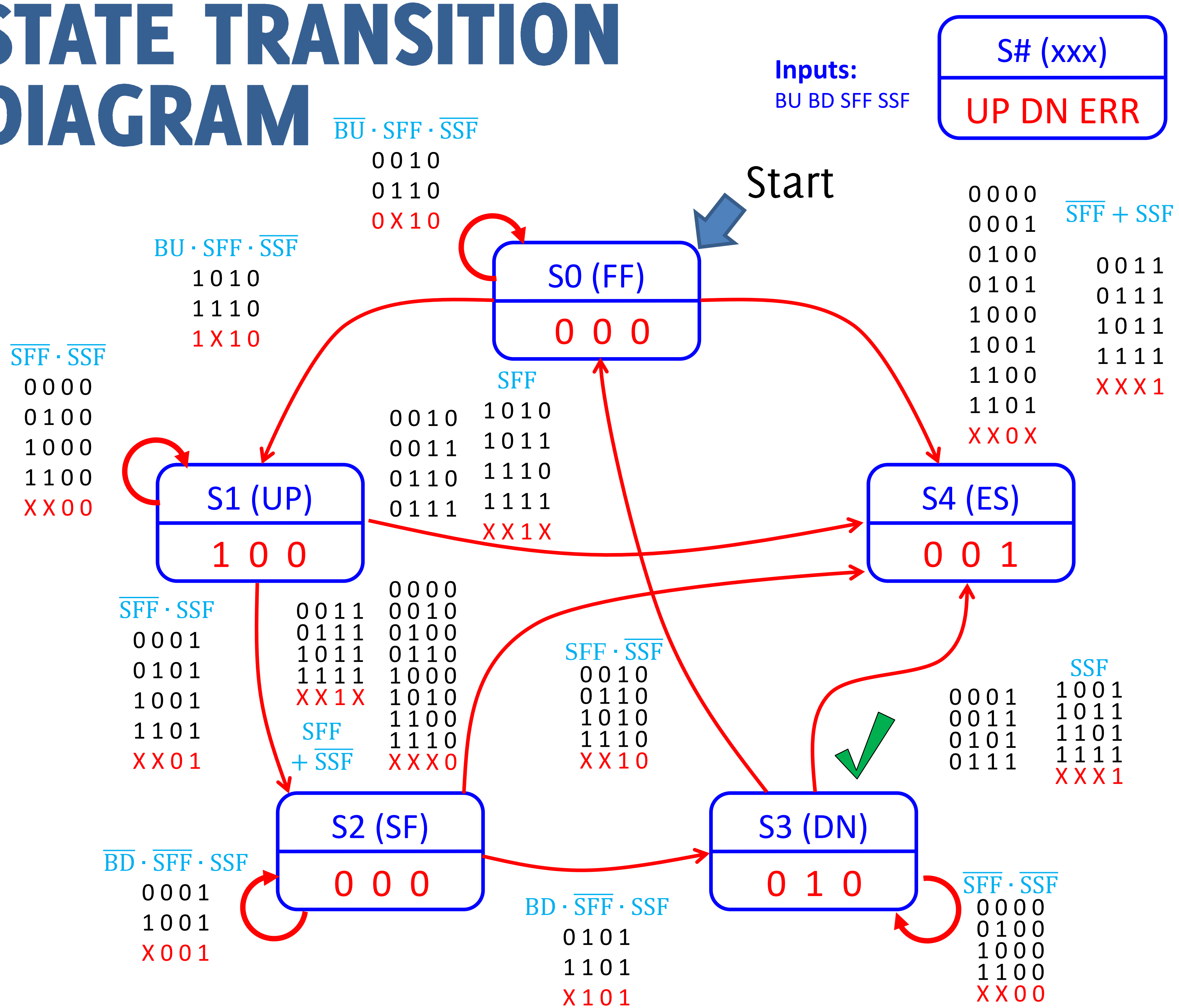
STATE TRANSITION DIAGRAM



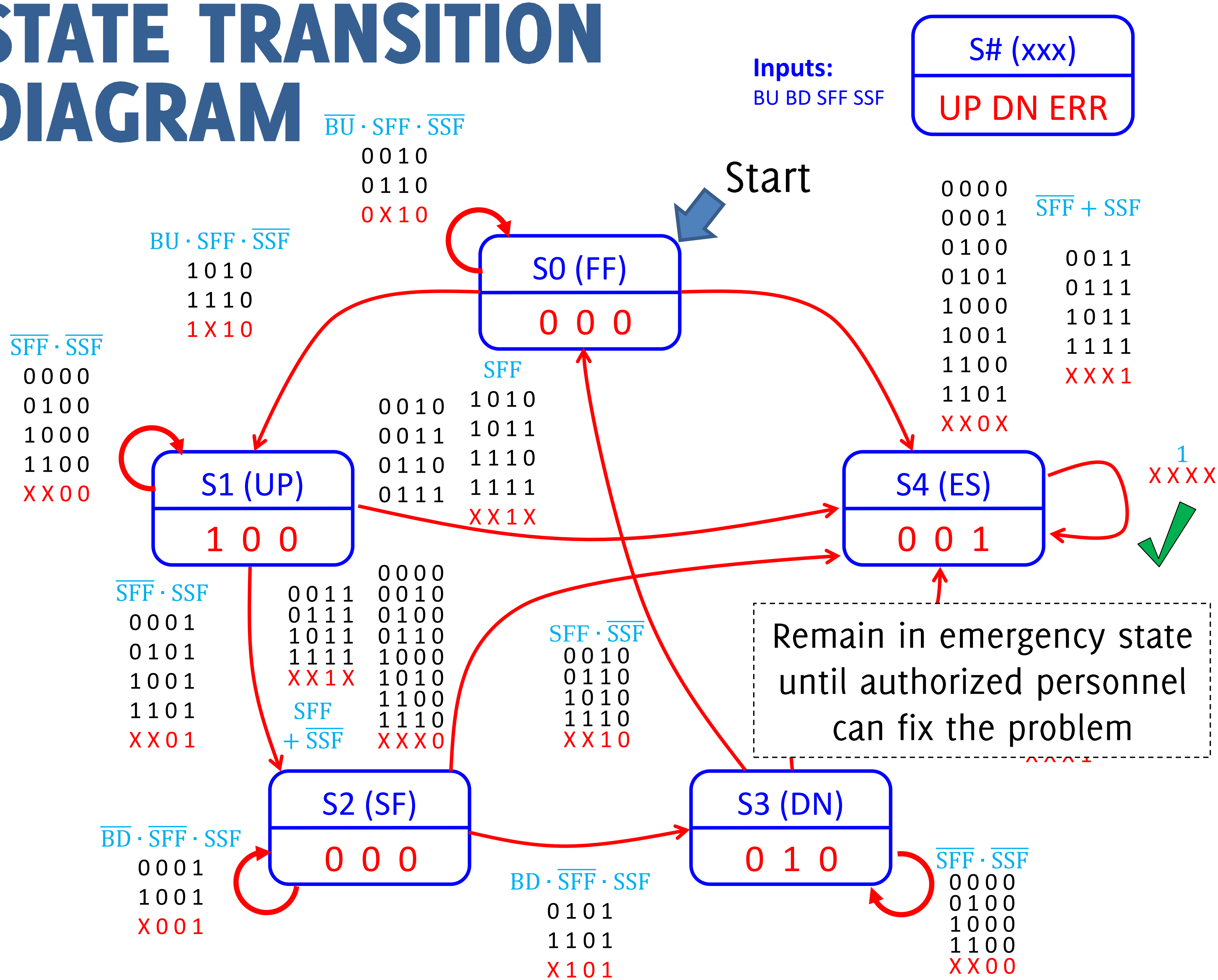
STATE TRANSITION DIAGRAM



STATE TRANSITION DIAGRAM



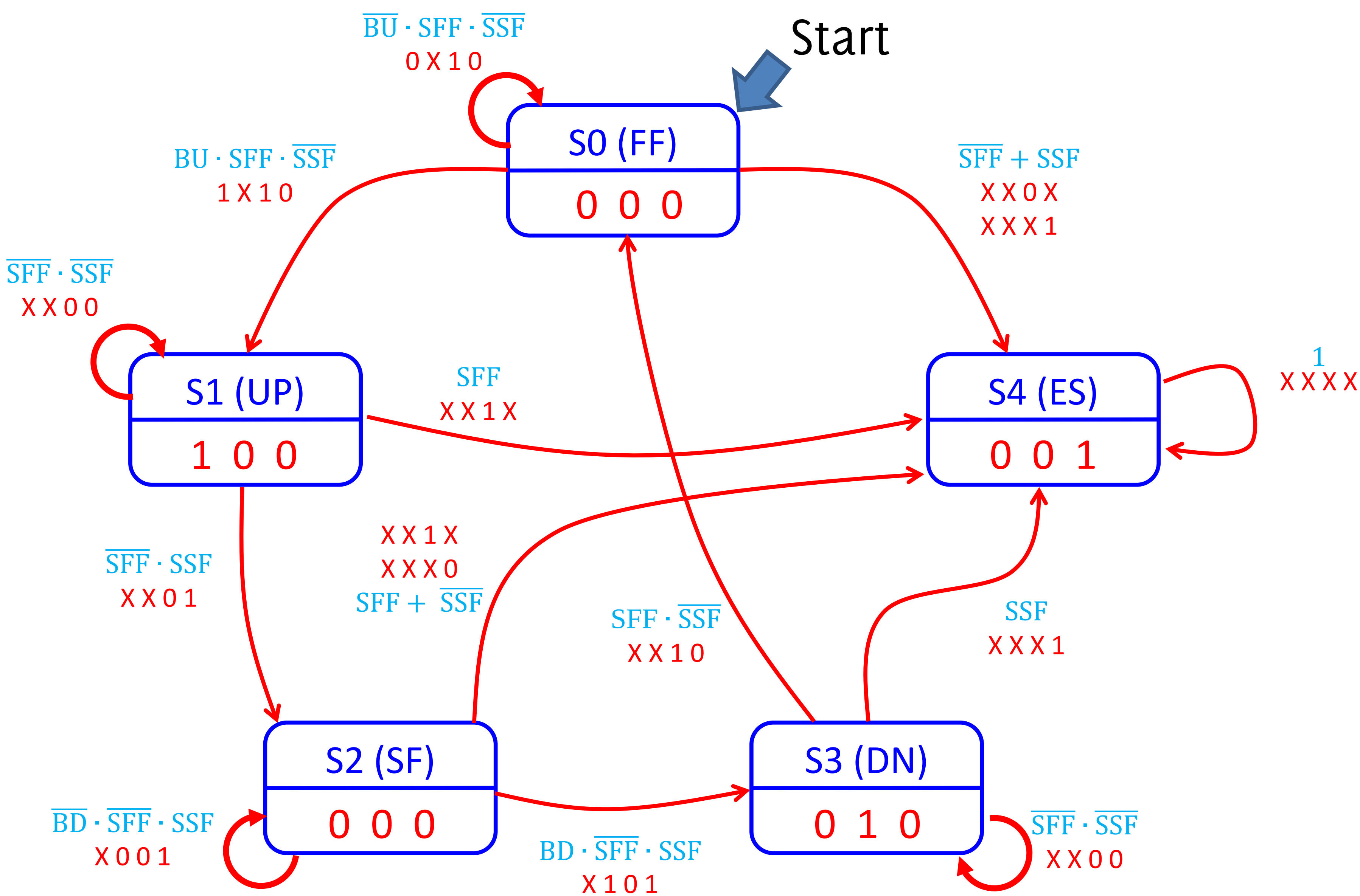
STATE TRANSITION DIAGRAM



STATE TRANSITION DIAGRAM

Inputs:
BU BD SFF SSF

S# (xxx)
UP DN ERR



STATE TRANSITION TABLE

Also known as a ***characteristic*** table

Only 1/5 of entire characteristic table shown here. Need all input combinations for all five states!

Inputs				Current	Next	Outputs		
BU	BD	SFF	SSF	State	State	UP	DN	ERR
0	0	0	0	S0	S4	0	0	0
0	0	0	1	S0	S4	0	0	0
0	0	1	0	S0	S0	0	0	0
0	0	1	1	S0	S4	0	0	0
0	1	0	0	S0	S4	0	0	0
0	1	0	1	S0	S4	0	0	0
0	1	1	0	S0	S0	0	0	0
0	1	1	1	S0	S4	0	0	0
1	0	0	0	S0	S4	0	0	0
1	0	0	1	S0	S4	0	0	0
1	0	1	0	S0	S1	0	0	0
1	0	1	1	S0	S4	0	0	0
1	1	0	0	S0	S4	0	0	0
1	1	0	1	S0	S4	0	0	0
1	1	1	0	S0	S1	0	0	0
1	1	1	1	S0	S4	0	0	0

⋮

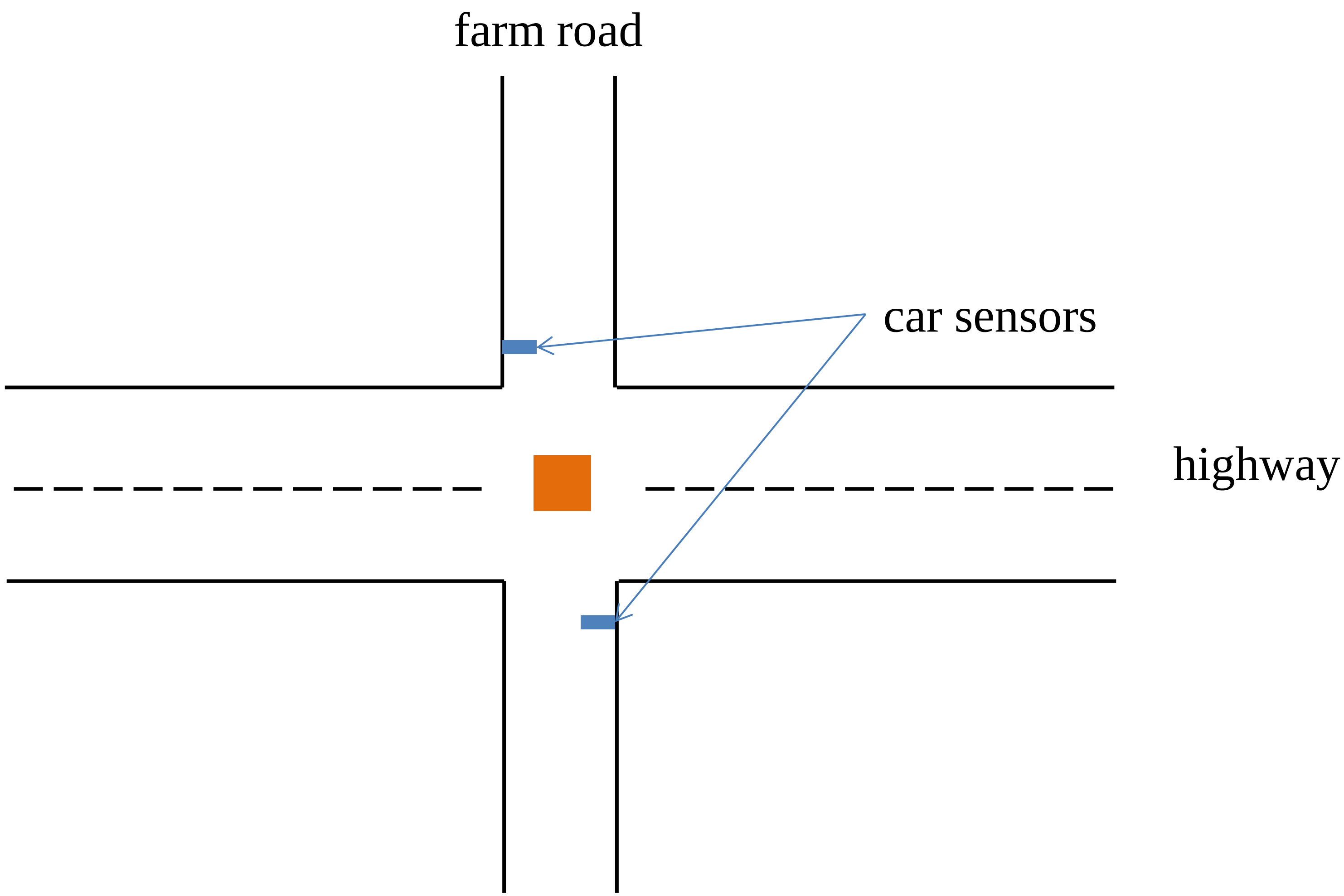
NEXT STATE TABLE

	Next State															
Current	**Inputs (BU BD SFF SSF)**															
State	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
S0	S4	S4	S0	S4	S4	S4	S0	S4	S4	S4	S1	S4	S4	S4	S1	S4
S1	S1	S2	S4	S4	S1	S2	S4	S4	S1	S2	S4	S4	S1	S2	S4	S4
S2	S4	S2	S4	S4	S4	S3	S4	S4	S4	S2	S4	S4	S4	S3	S4	S4
S3	S3	S4	S0	S4	S3	S4	S0	S4	S3	S4	S0	S4	S3	S4	S0	S4
S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4	S4

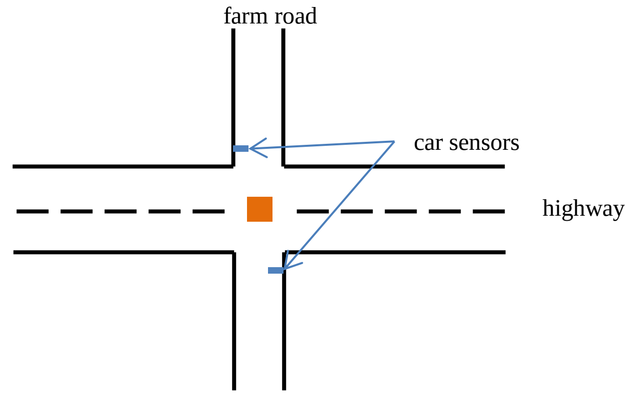
Also known as two-dimensional state table

TRAFFIC LIGHT EXAMPLE

A busy highway is intersected by a infrequently used farm road. Detectors C sense the presence of cars on the farm road.



PROBLEM STATEMENT

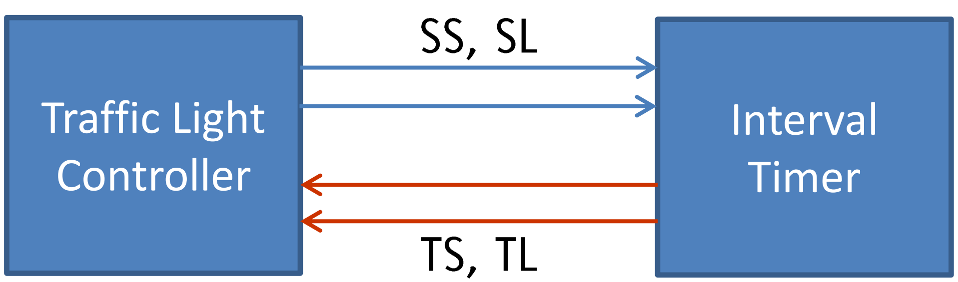


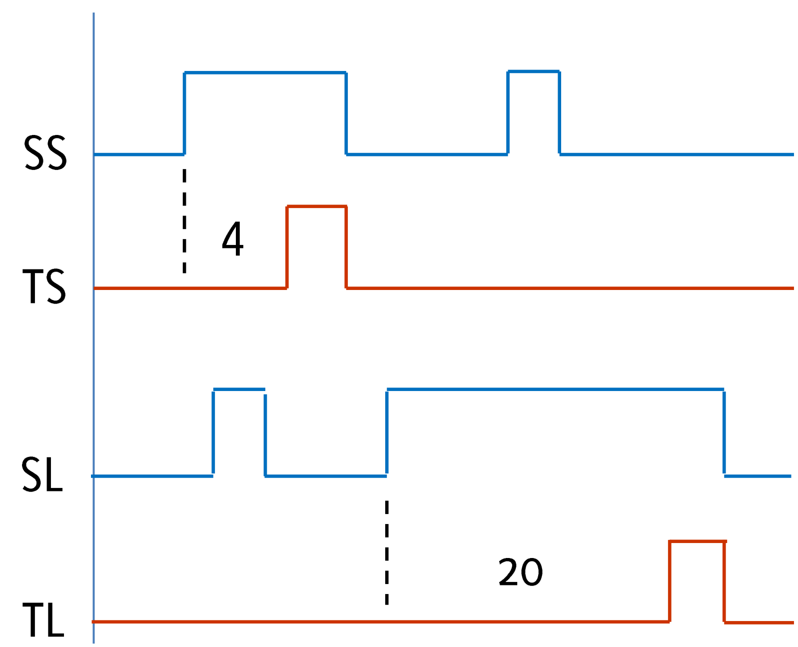
Desired operation

- With no car on farm road, lights remain Green in highway direction
- If vehicle detected on farm road, highway lights go from Green to Yellow to Red, allowing the farm road lights to become Green
- Farm road lights stay Green only as long as a farm road car is detected but never longer than a set interval (say, 20 seconds)
- When farm road traffic is gone, or time has expired, the farm lights transition from Green to Yellow to Red, allowing the highway light to return to Green
- Even if farm road vehicles are waiting, the highway gets a minimum amount of time with Green light on (say, 20 seconds)

TIMER INFORMATION

- Two interval timers are available, one with a short time interval (4 sec) and one with a long time interval (20 sec).
- Short timing is initiated when the short start signal SS is activated (SS=1). Signal TS goes high (TS=1) after SS remains activated for at least 4 seconds, and is reset (TS=0) when SS goes low (SS=0).
- Long timing is initiated when the long start signal SL is activated (SL=1). Signal TL goes high (TL=1) after SL remains activated for at least 20 seconds, and is reset (TL=0) when SL goes low (SS=0).





INPUTS AND OUTPUTS (IGNORING TIMERS)

Inputs

CS = 1: car detected on farm road

CS = 0: no cars on farm road

Outputs

HG = 1: highway green light on

HG = 0: highway green light off

HY = 1: highway yellow light on

HY = 0: highway yellow light off

HR = 1: highway red light on

HR = 0: highway red light off

FG = 1: farm road green light on

FG = 0: farm road green light off

FY = 1: farm road yellow light on

FY = 0: farm road yellow light off

FR = 1: farm road red light on

FR = 0: farm road red light off

INPUTS AND OUTPUTS

Inputs

CS = 1: car detected on farm road
CS = 0: no cars on farm road
TS = 0: short timer not expired
TS = 1: short timer expired
TL = 0: long timer not expired
TL = 1: long timer expired

Outputs

HG = 1: highway green light on
HG = 0: highway green light off
HY = 1: highway yellow light on
HY = 0: highway yellow light off
HR = 1: highway red light on
HR = 0: highway red light off
FG = 1: farm road green light on
FG = 0: farm road green light off
FY = 1: farm road yellow light on
FY = 0: farm road yellow light off
FR = 1: farm road red light on
FR = 0: farm road red light off
SS = 1: short timer is active
SS = 0: reset short timer
SL = 1: long timer is active
SL = 0: reset long timer

REDUNDANT INPUTS?

Inputs

CS = 1: car detected on farm road
CS = 0: no cars on farm road
TS = 0: short timer not expired
TS = 1: short timer expired
TL = 0: long timer not expired
TL = 1: long timer expired

Outputs

HG = 1: highway green light on
HG = 0: highway green light off
HY = 1: highway yellow light on
HY = 0: highway yellow light off
HR = 1: highway red light on
HR = 0: highway red light off
FG = 1: farm road green light on
FG = 0: farm road green light off
FY = 1: farm road yellow light on
FY = 0: farm road yellow light off
FR = 1: farm road red light on
FR = 0: farm road red light off
SS = 1: short timer is active
SS = 0: reset short timer
SL = 1: long timer is active
SL = 0: reset long timer

STATES?

STATES?

Possible States

SP0: highway green light on

SP1: highway yellow light on

SP2: highway red light on

SP3: farm road green light on

SP4: farm road yellow light on

SP5: farm road red light on

REDUNDANT STATES?

Possible States

SP0: highway green light on

SP1: highway yellow light on

SP2: highway red light on

SP3: farm road green light on

SP4: farm road yellow light on

SP5: farm road red light on

REDUNDANT STATES?

Possible States

SP0: highway green light on

SP1: highway yellow light on

SP2: highway red light on

SP3: farm road green light on

SP4: farm road yellow light on

SP5: farm road red light on

REDUNDANT STATES?

Possible States

SP0: highway green light on

SP1: highway yellow light on

SP2: highway red light on

SP3: farm road green light on

SP4: farm road yellow light on

SP5: farm road red light on

States

S0: highway green light on (HG)

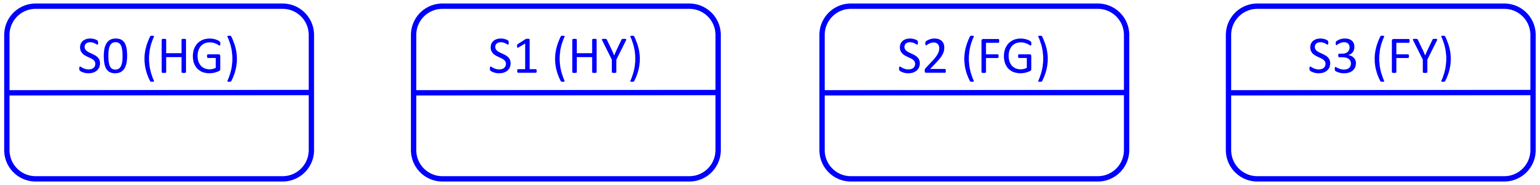
S1: highway yellow light on (HY)

S2: farm road green light on (FG)

S3: farm road yellow light on (FY)

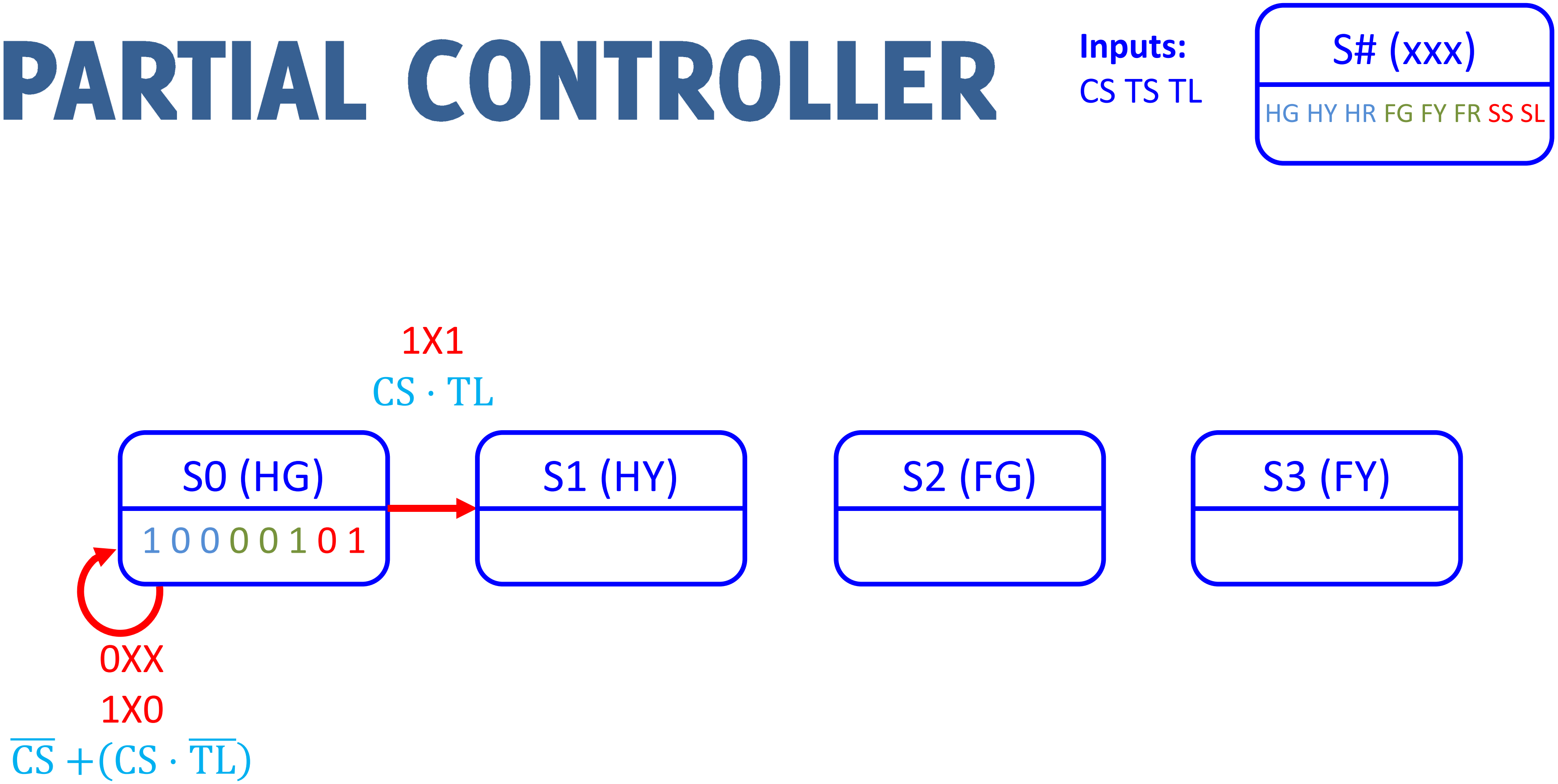
PARTIAL CONTROLLER

Inputs:
CS TS TL



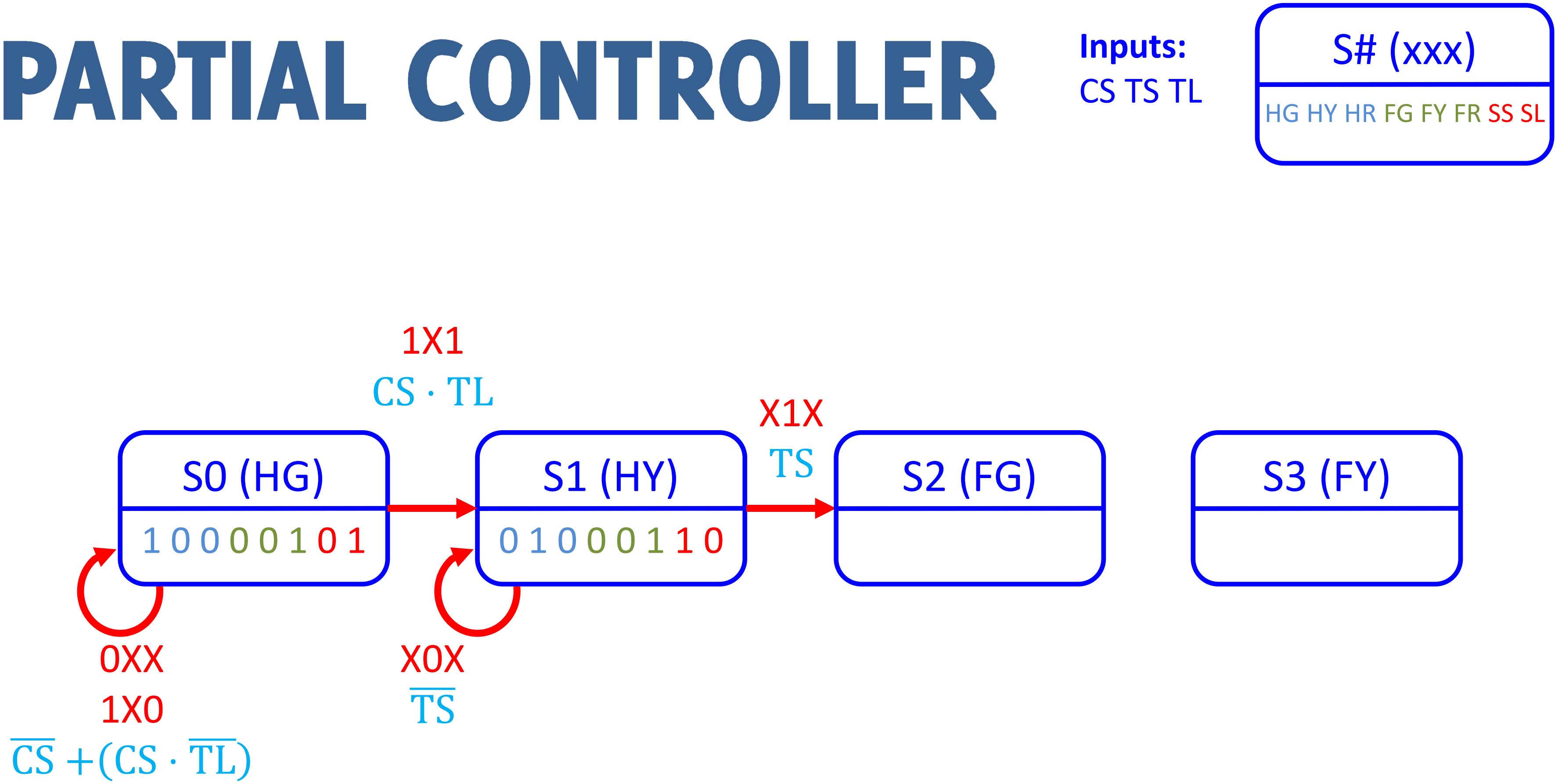
PARTIAL CONTROLLER

Inputs:
CS TS TL



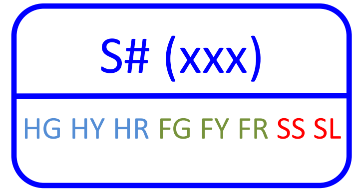
PARTIAL CONTROLLER

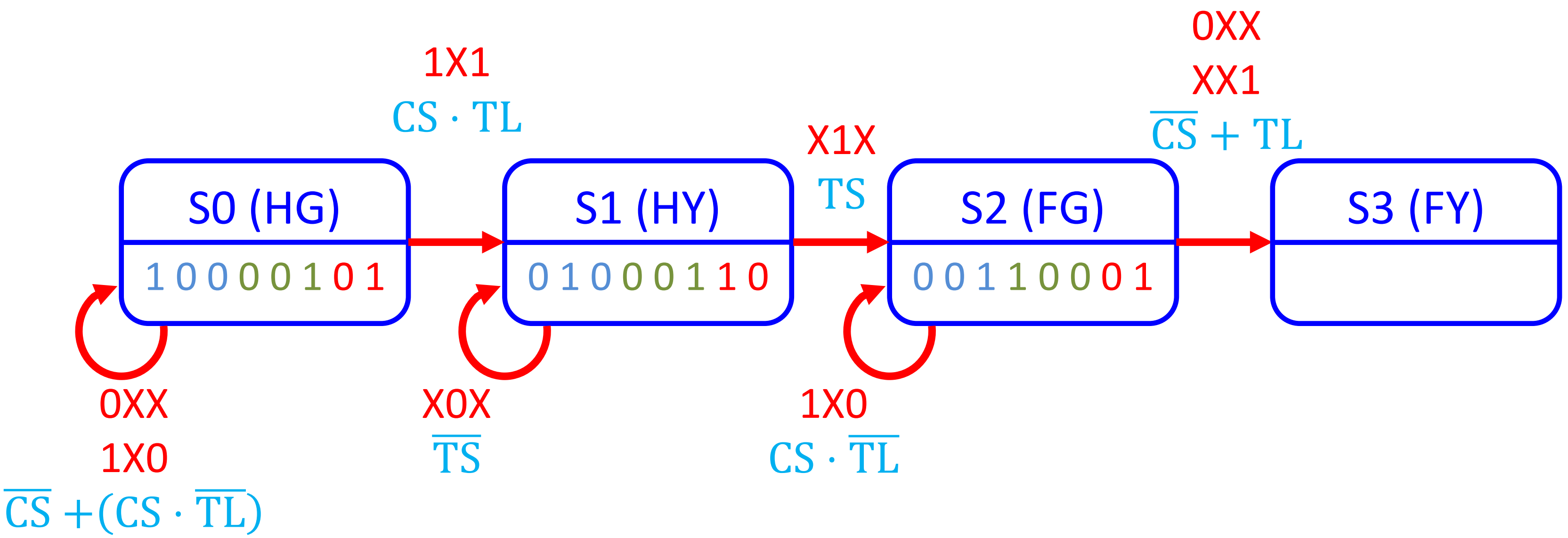
Inputs:
CS TS TL



PARTIAL CONTROLLER

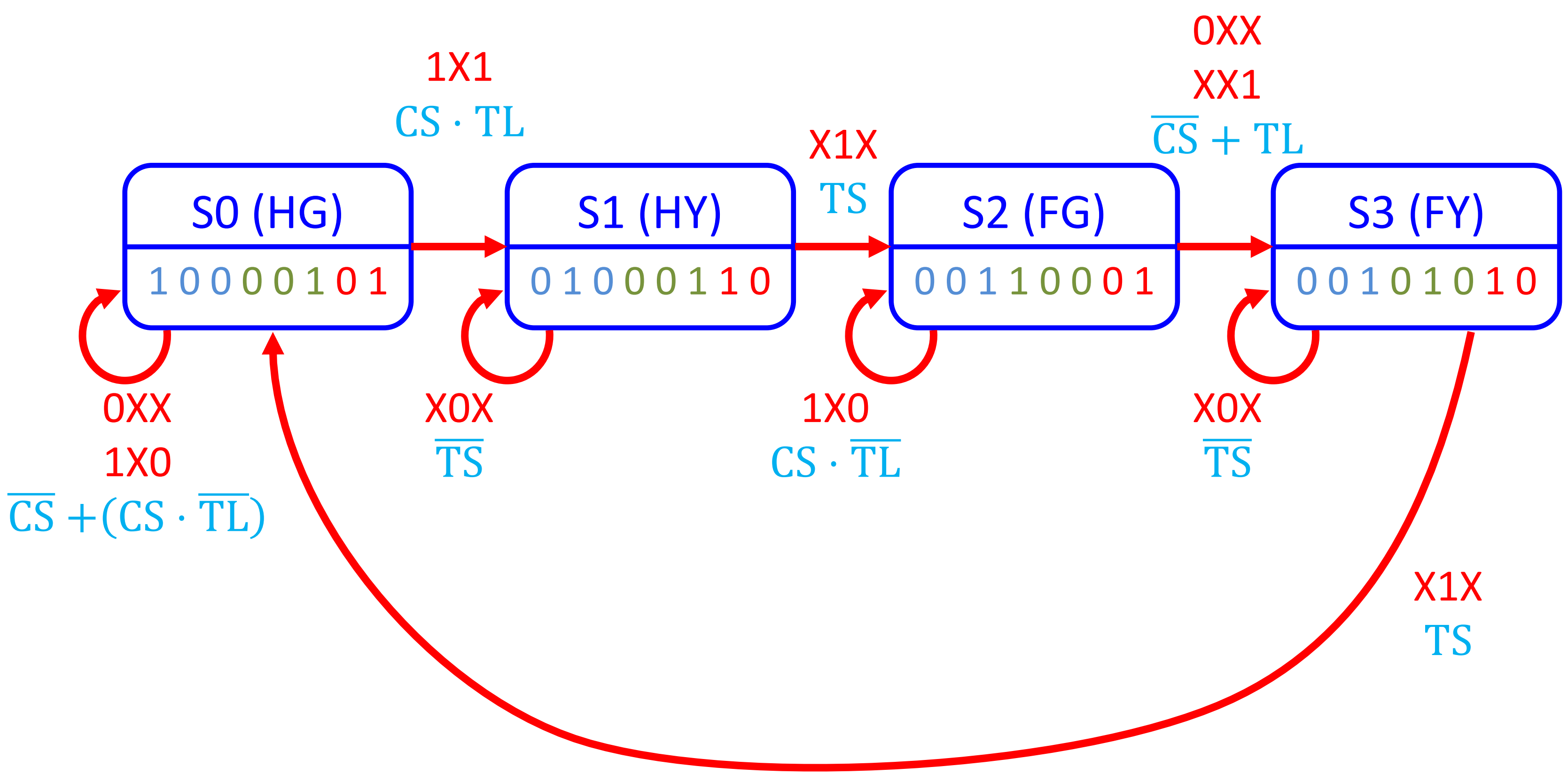
Inputs:
CS TS TL





FINAL CONTROLLER

Inputs:
CS TS TL



STATE TRANSITION TABLE

Also known as
characteristic table

Inputs			Current	Next	Current Outputs							
CS	TS	TL	State	State	HG	HY	HR	FG	FY	FR	SS	SL
0	0	0	HG	HG	1	0	0	0	0	1	0	1
0	0	1	HG	HG	1	0	0	0	0	1	0	1
0	1	0	HG	HG	1	0	0	0	0	1	0	1
0	1	1	HG	HG	1	0	0	0	0	1	0	1
1	0	0	HG	HG	1	0	0	0	0	1	0	1
1	0	1	HG	HY	1	0	0	0	0	1	0	1
1	1	0	HG	HG	1	0	0	0	0	1	0	1
1	1	1	HG	HY	1	0	0	0	0	1	0	1
0	0	0	HY	HY	0	1	0	0	0	1	1	0
0	0	1	HY	HY	0	1	0	0	0	1	1	0
0	1	0	HY	FG	0	1	0	0	0	1	1	0
0	1	1	HY	FG	0	1	0	0	0	1	1	0
1	0	0	HY	HY	0	1	0	0	0	1	1	0
1	0	1	HY	HY	0	1	0	0	0	1	1	0
1	1	0	HY	FG	0	1	0	0	0	1	1	0
1	1	1	HY	FG	0	1	0	0	0	1	1	0
0	0	0	FG	FY	0	0	1	1	0	0	0	1
0	0	1	FG	FY	0	0	1	1	0	0	0	1
0	1	0	FG	FY	0	0	1	1	0	0	0	1
0	1	1	FG	FY	0	0	1	1	0	0	0	1
1	0	0	FG	FG	0	0	1	1	0	0	0	1
1	0	1	FG	FY	0	0	1	1	0	0	0	1
1	1	0	FG	FG	0	0	1	1	0	0	0	1
1	1	1	FG	FY	0	0	1	1	0	0	0	1
0	0	0	FY	FY	0	0	1	0	1	0	1	0
0	0	1	FY	FY	0	0	1	0	1	0	1	0
0	1	0	FY	HG	0	0	1	0	1	0	1	0
0	1	1	FY	HG	0	0	1	0	1	0	1	0
1	0	0	FY	FY	0	0	1	0	1	0	1	0
1	0	1	FY	FY	0	0	1	0	1	0	1	0
1	1	0	FY	HG	0	0	1	0	1	0	1	0
1	1	1	FY	HG	0	0	1	0	1	0	1	0

NEXT STATE TABLE

	Next State							
Current	Inputs (CS TS TL)							
State	000	001	010	011	100	101	110	111
HG	HG	HG	HG	HG	HG	HY	HG	HY
HY	HY	HY	FG	FG	HY	HY	FG	FG
FG	FY	FY	FY	FY	FG	FY	FG	FY
FY	FY	FY	HG	HG	FY	FY	HG	HG

Also known as two-dimensional state table