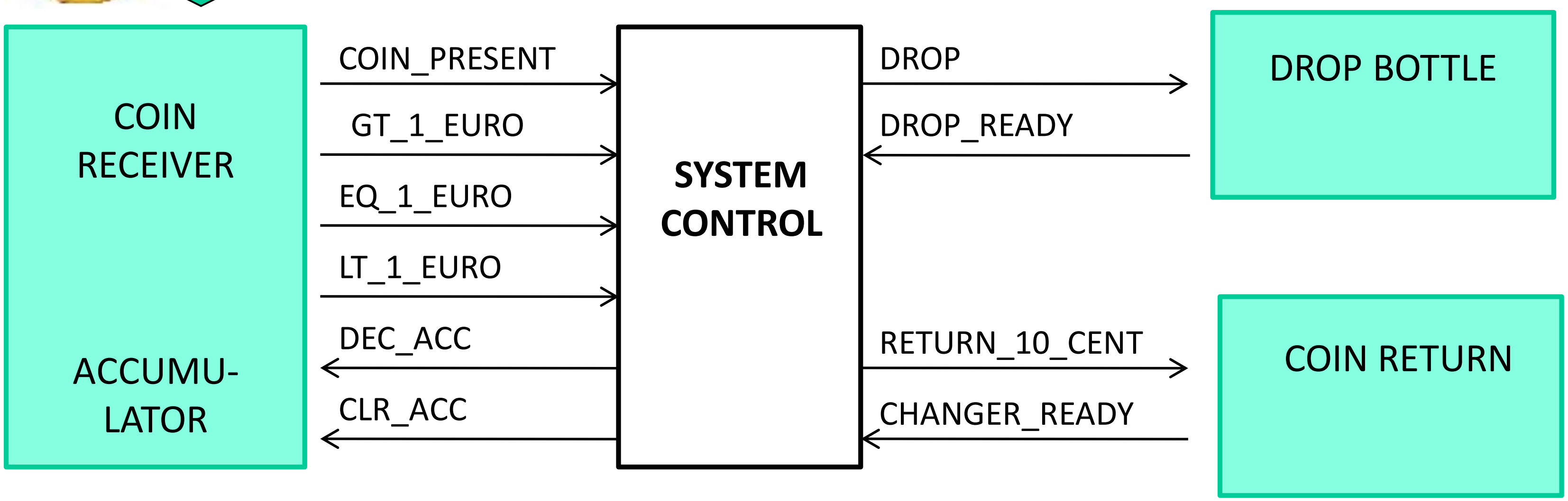
# Vending machine

## Design-exempel by *Ingo Sander*

William Sandqvist william@kth.se

# System Control


We will design the block,
**System Control**

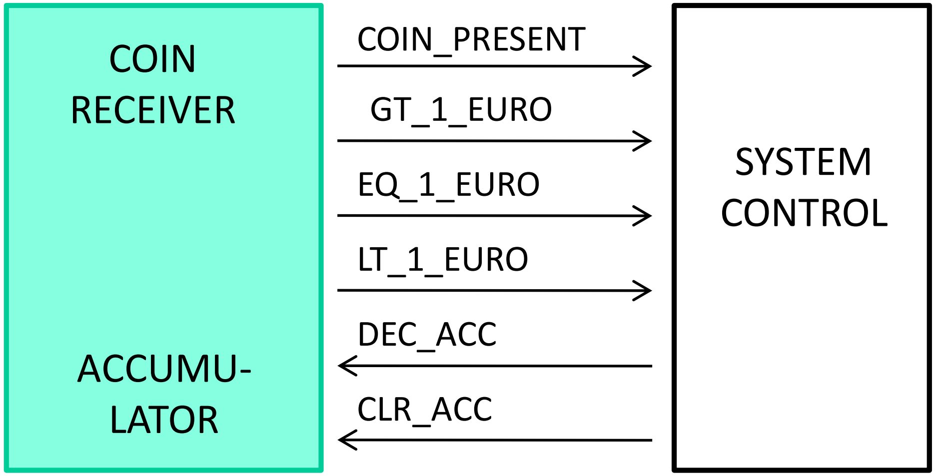



Return only 10 cents coin.

William Sandqvist william@kth.se

# Coin Reciever

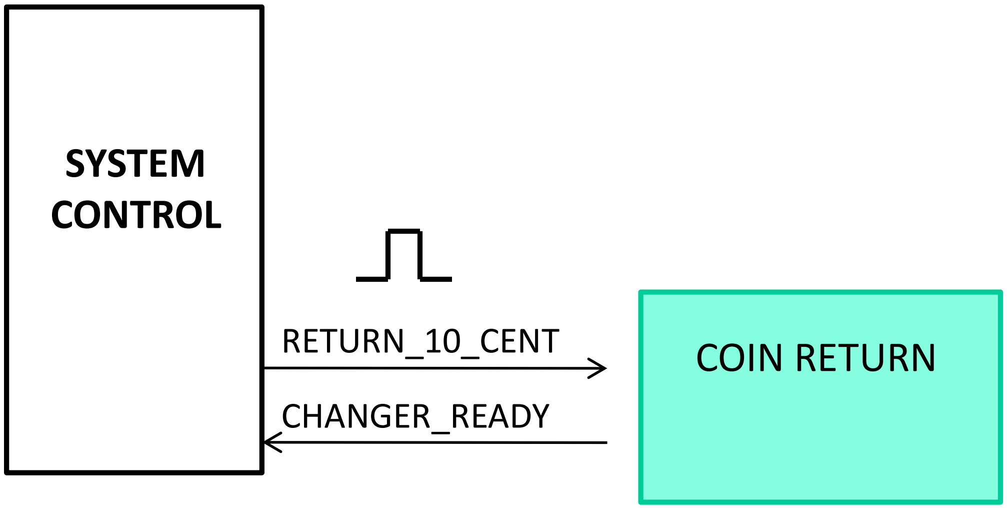
The **System Control** unit controls a number of subsystems from other suppliers. **Coin Reciever**. **Drop Bottle**. **Coin Return**.




An ACCUMULATOR counts up the amount of paid coins.

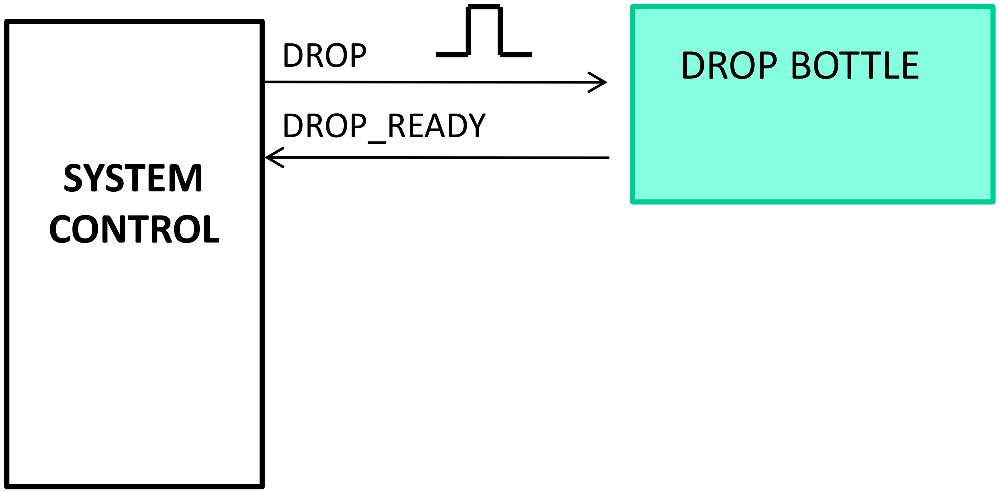
- Signal COIN_PRESENT indicates that there are coins and the ”amount” is indicated by the signals GT_1_EURO, EQ_1_EURO, LT_1_EURO.
- With the signals DEC_ACC and CLR_ACC the systemcontrol unit can reduce the amount by 10 cents, or reset the ACCUMULATOR.

# Coin Return



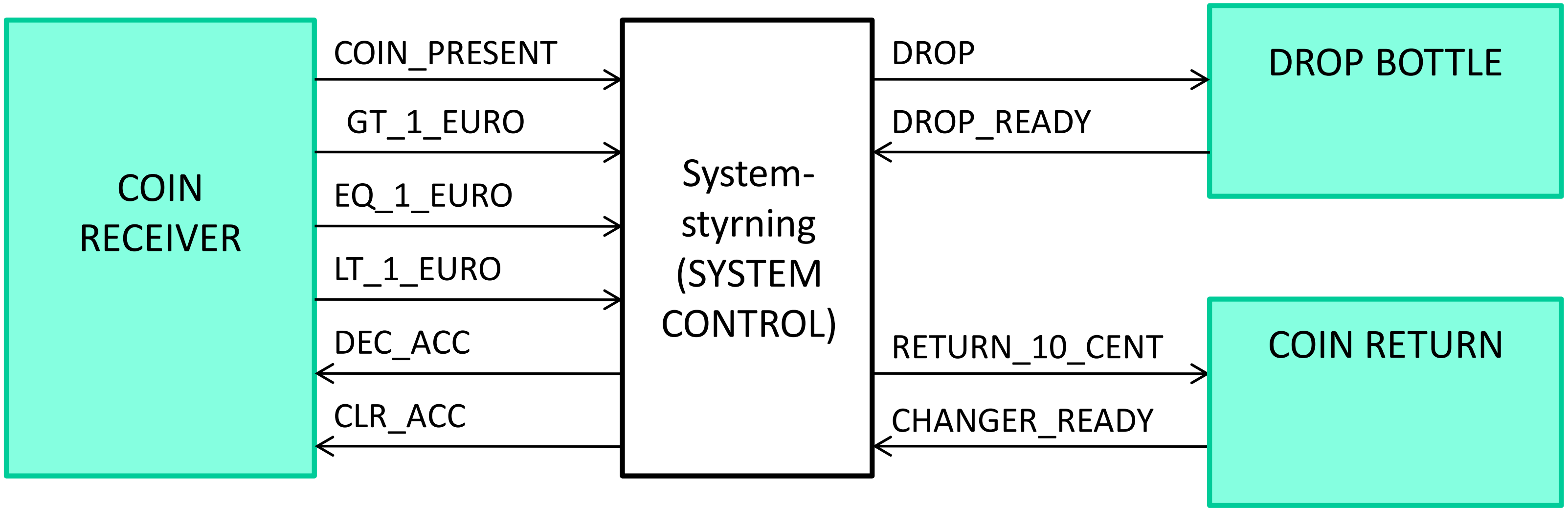

With a pulse on RETURN_10_CENT the coin return unit will eject 10 cent, and signal CHANGER_READY when this is done and the unit is ready for the next command.

# Drop Bottle



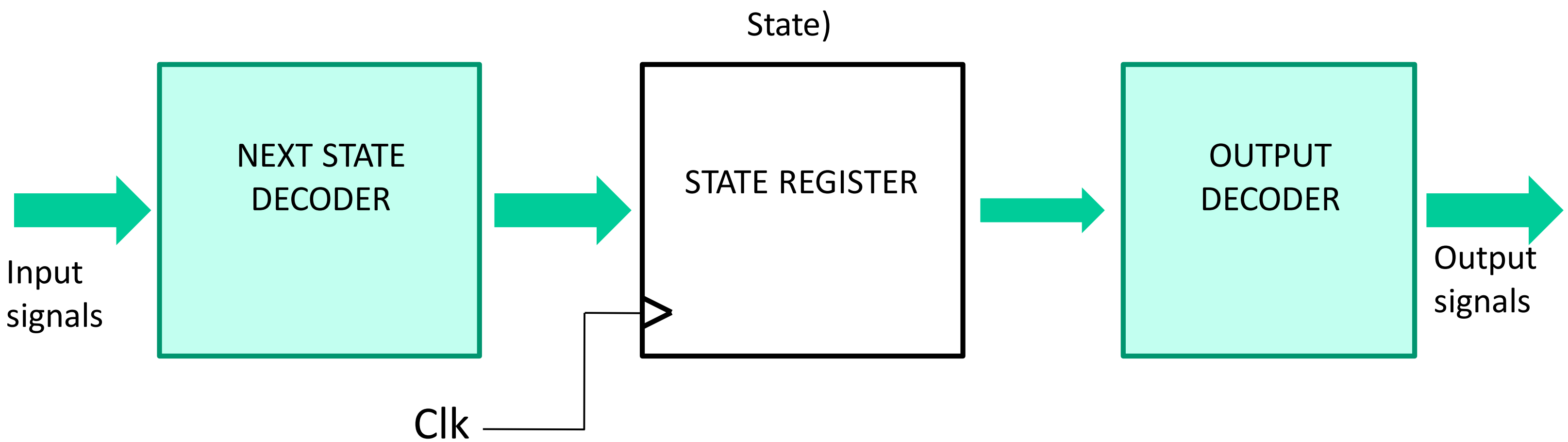

With a pulse on DROP the drop bottle unit will eject a bottle, and signals DROP_READY when this is done and the unit is ready for the next command.

# Block diagram



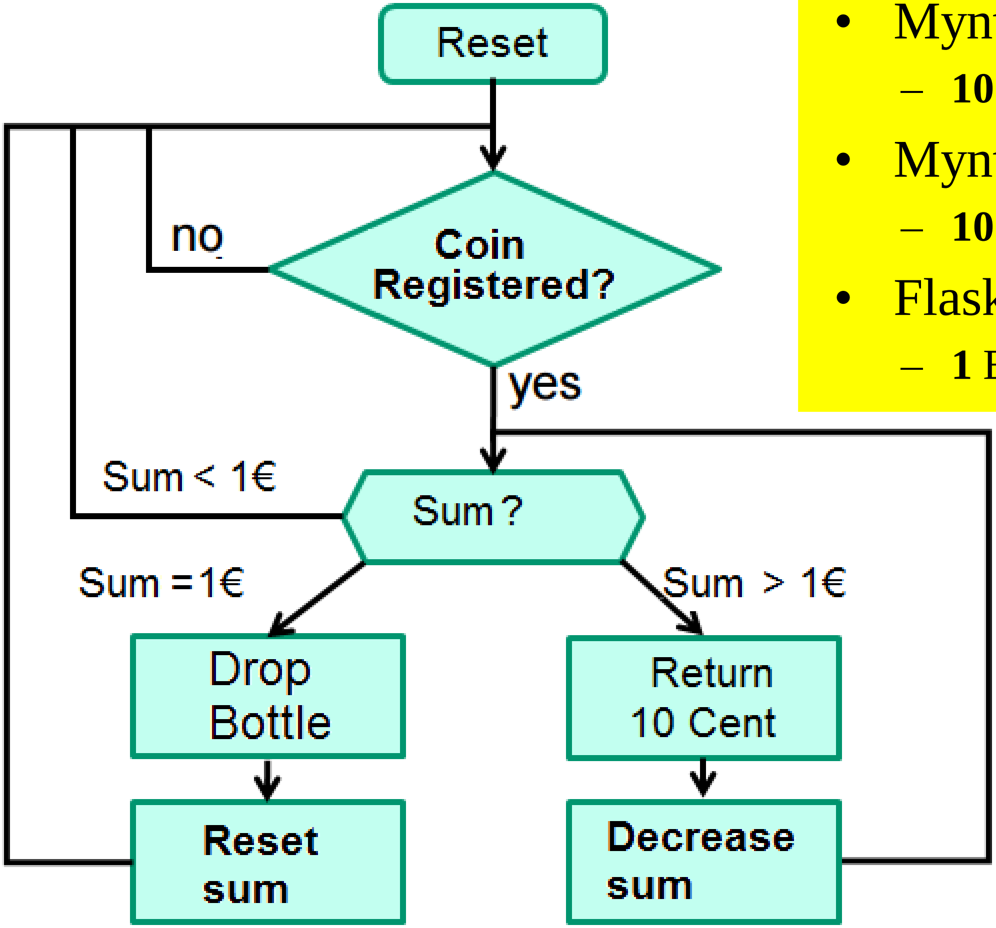

- Signal properties
  - DROP_READY is active for a clockperiod after the bottle is ejected
  - CHANGER_READY is active for a clockperiod after a 10 Cent coin is ejected
    - Because of the mechanical properties the following signals are active and inactive for several clock periods:
      - COIN_PRESENT (active for several clockperiods after the coin is inserted)
      - DROP_READY (active for several clockperiods at bottle ejection)
      - CHANGER_READY (inactive for several clockperiods at coin ejection )

# Use a Moore-machine

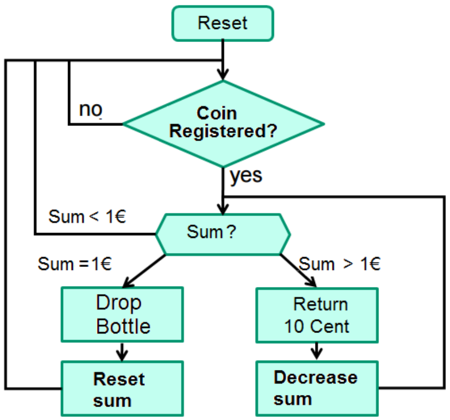



- Construction of a state machine for the controller of a vending machine
- Assumptions
  - Moore-Machine
  - - Stateregister implemented with D-flip-flops

# Function diagram for vending machine

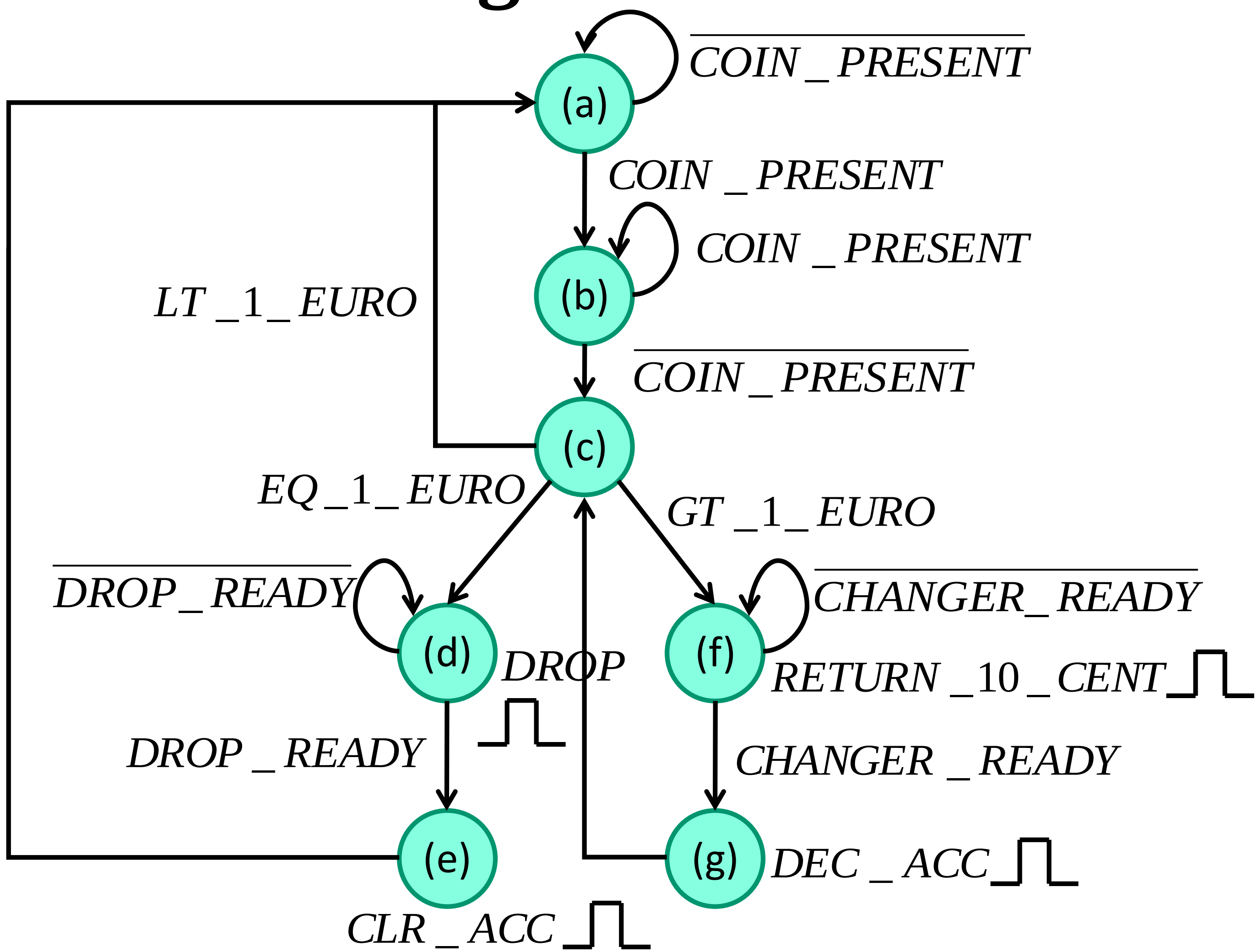



- Myntinkast
  - **10** Cent, **50** Cent, **1** Euro
- Myntutkast
  - **10** Cent
- Flaskpris
  - **1** Euro

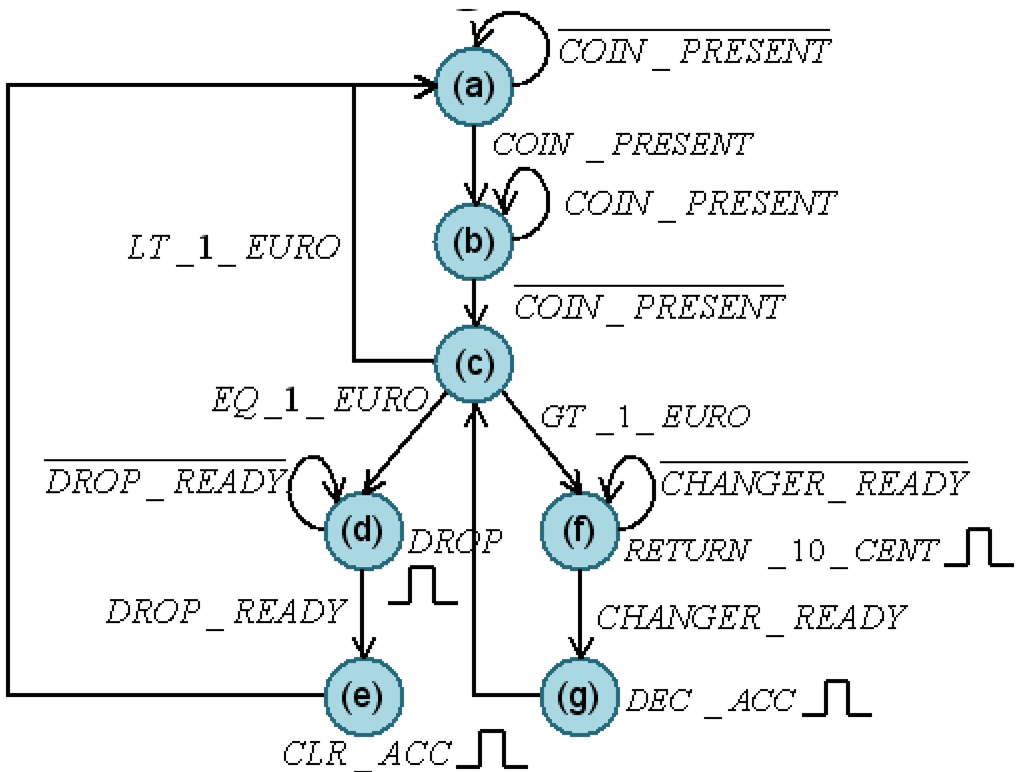



# State diagram

We draw a state diagram from the functional diagram :



# State diagram

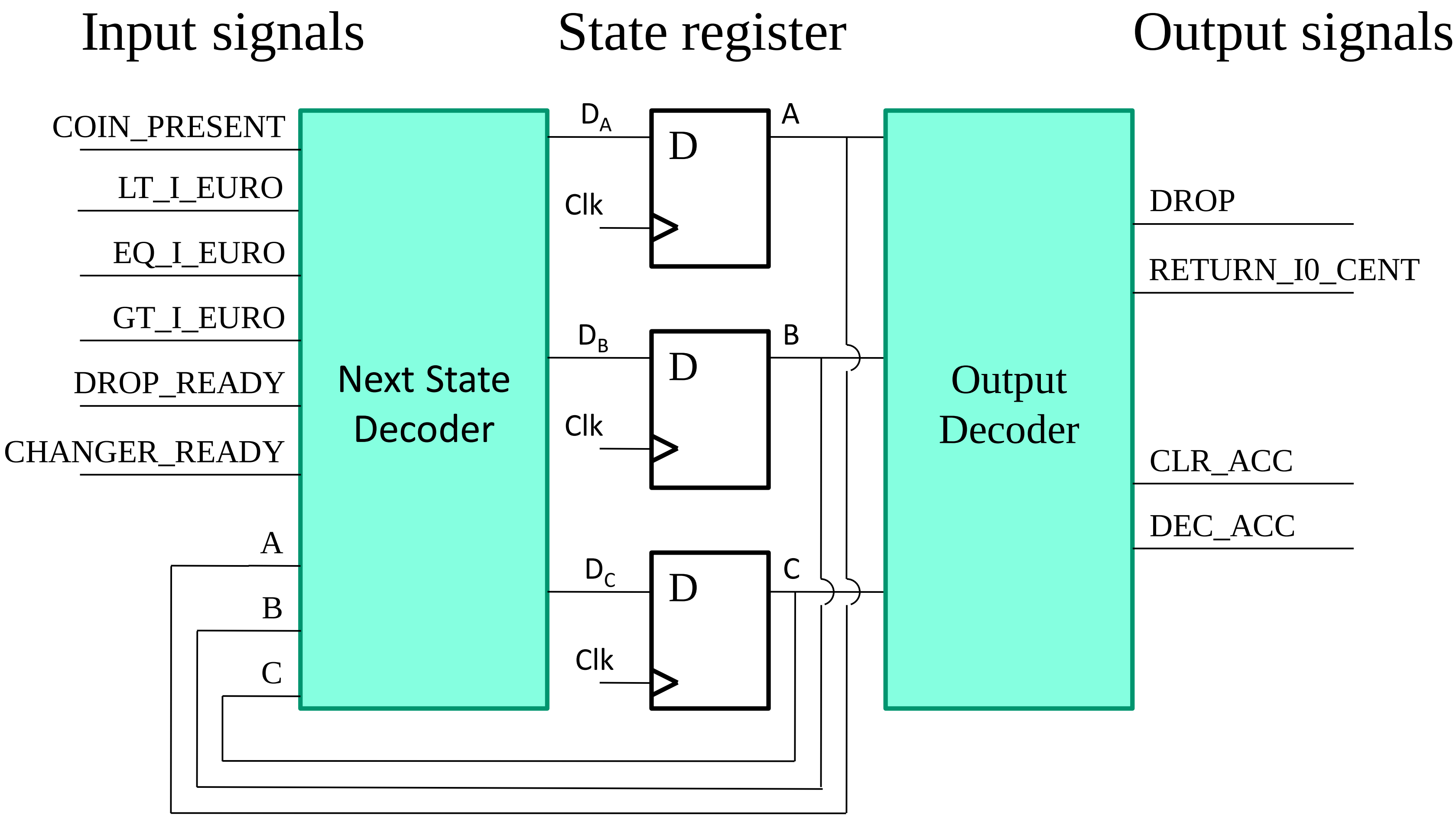



- (a) Wait for coin input
- (b) Register coin?
- (c) Coin is registered (3 cases)

- (d) Eject bottle
- (e) Reset sum
- (f) Return 10 Cent
- (g) Decrease sum 10 Cent

William Sandqvist william@kth.se

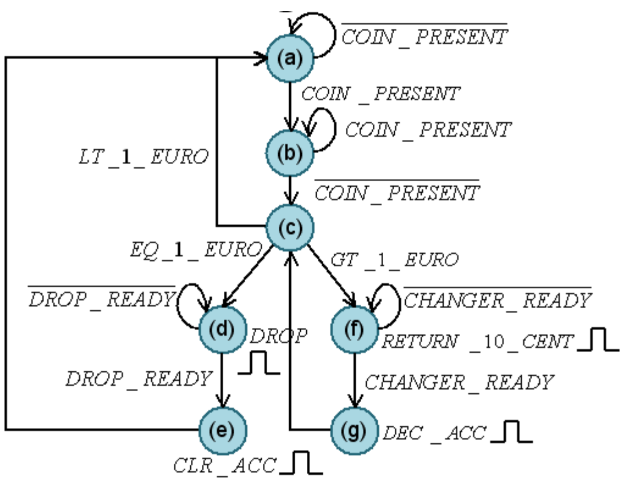
# Block schematic



# State encoding

Idea: Let the states that are close together in the state diagram have codes with unit distance.

(b) next to (c)
(d) next to (e)
(a) next to (b)
(f) next to (g)




7 states 3 state variables A, B, C are needed

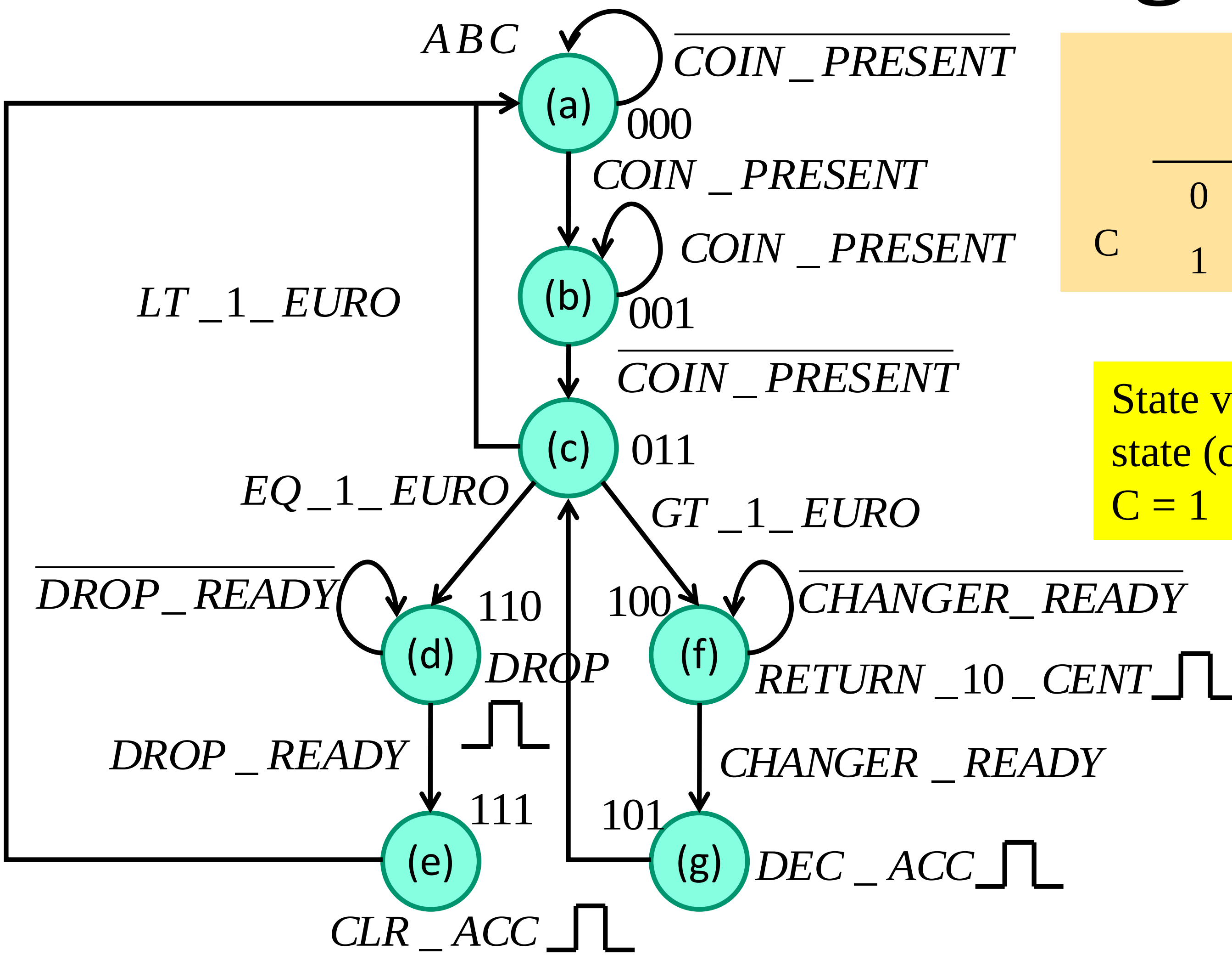
| | | AB | | | |
|---|---|---|---|---|---|
| | | 00 | 01 | 11 | 10 |
| C | 0 | a | Ø | d | f |
| | 1 | b | c | e | g |

(Ø = don't care)

*The number of inputs is large, 6, total there may be nine variables in Karnaugh maps ???*

William Sandqvist william@kth.se

# State encoding




| | | AB | | | |
|---|---|---|---|---|---|
| | | 00 | 01 | 11 | 10 |
| C | 0 | a | Ø | d | f |
| | 1 | b | c | e | g |

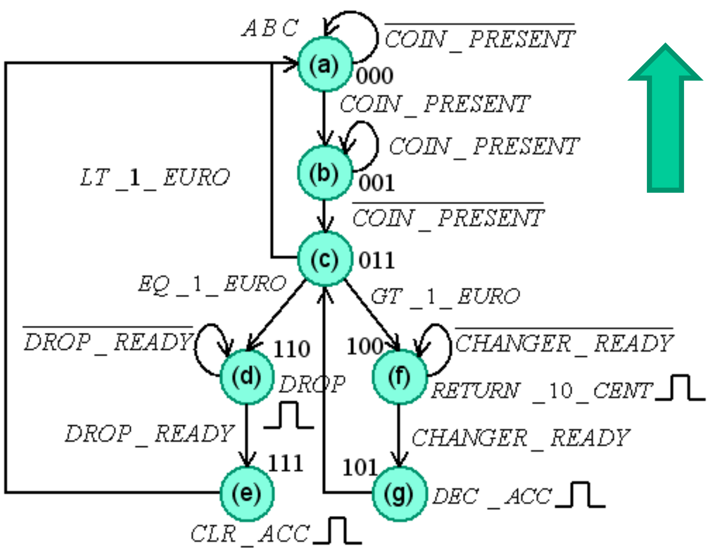
(Ø = don’t care)

State variables ABC, eg. in state (c) is A = 0, B = 1 and C = 1

# Coded state table?

$$A^+B^+C^+ = f(A\,B\,C, CP, DR, CR, GT, LT, EQ)$$

| $C$ \ $AB$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $(a):$ $\overline{CP} \rightarrow 000\ (a)$ <br> $CP \rightarrow 001\ (b)$ | $\Phi : \Phi$ | $(d):$ $\overline{DR} \rightarrow 110\ (d)$ <br> $DR \rightarrow 111\ (e)$ | $(f):$ $\overline{CR} \rightarrow 100\ (f)$ <br> $CR \rightarrow 101\ (g)$ |
| 1 | $(b):$ $CP \rightarrow 001\ (b)$ <br> $\overline{CP} \rightarrow 011\ (c)$ | $(c):$ $GT \rightarrow 100\ (f)$ <br> $EQ \rightarrow 110\ (d)$ <br> $LT \rightarrow 000\ (a)$ | $(e): \rightarrow 000\ (a)$ | $(g): \rightarrow 011\ (c)$ |




From the state diagram, you can set up the following coded state table.

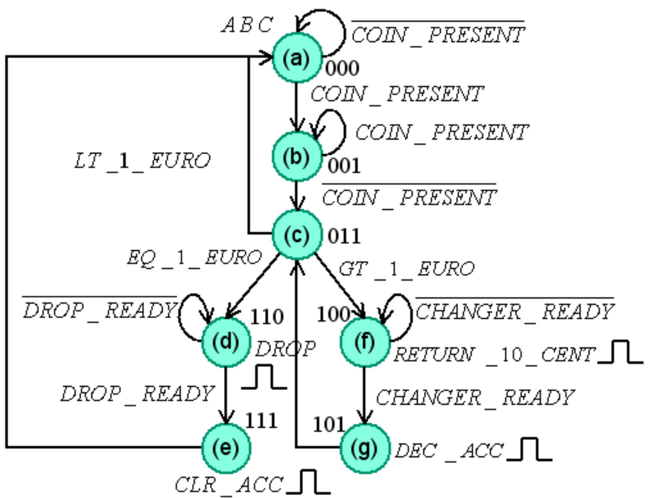
How do we avoid the complexity of **nine** variables?

# Variable-Entered Mapping (VEM)

Variable-Entered Mapping can be helpful when you need Karnaugh diagrams with many variables. You write functional expressions in the Karnaugh map.

$$A^+B^+C^+ = f(ABC, CP, DR, CR, GT, LT, EQ)$$

| $C$ \ $AB$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $(a): \overline{CP}\to 000\,(a)$; $CP\to 001\,(b)$ | $\Phi:\Phi$ | $(d): \overline{DR}\to 110\,(d)$; $DR\to 111\,(e)$ | $(f): \overline{CR}\to 100\,(f)$; $CR\to 101\,(g)$ |
| 1 | $(b): CP\to 001\,(b)$; $\overline{CP}\to 011\,(c)$ | $(c): GT\to 100\,(f)$; $EQ\to 110\,(d)$; $LT\to 000\,(a)$ | $(e): \to 000\,(a)$ | $(g): \to 011\,(c)$ |




| $A^+$ $D_A$ | | AB | | | |
|---|---|---|---|---|---|
| | | 00 | 01 | 11 | 10 |
| C | 0 | 0 | Ø | 1 | 1 |
| | 1 | 0 | $\boxed{EQ + GT}$ | 0 | 0 |

*Variables*

# Next state - $D_A$

$$A^{+}B^{+}C^{+} = f(ABC, CP, DR, CR, GT, LT, EQ)$$

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $(a): \overline{CP} \to 000\ (a)$; $CP \to 001\ (b)$ | $\Phi : \Phi$ | $(d): \overline{DR} \to 110\ (d)$; $DR \to 111\ (e)$ | $(f): \overline{CR} \to 100\ (f)$; $CR \to 101\ (g)$ |
| 1 | $(b): CP \to 001\ (b)$; $\overline{CP} \to 011\ (c)$ | $(c): GT \to 100\ (f)$; $EQ \to 110\ (d)$; $LT \to 000\ (a)$ | $(e): \to 000\ (a)$ | $(g): \to 011\ (c)$ |

**$A^{+}$ $D_A$**

| C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 0 | Ø | 1 | 1 |
| 1 | 0 | EQ + GT | 0 | 0 |

*EQ* : EQ_1_EURO
*GT* : GT_1_EURO

$$A^{+} = D_A = \overline{A} \cdot B \cdot EQ + \overline{A} \cdot B \cdot GT + A \cdot \overline{C}$$

# Next state - $D_B$

$$A^+B^+C^+ = f\left(A\,B\,C, CP, DR, CR, GT, LT, EQ\right)$$

| $C$ \ $AB$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $(a): \begin{array}{l}\overline{CP} \to 000\ (a)\\ CP \to 001\ (b)\end{array}$ | $\Phi : \Phi$ | $(d): \begin{array}{l}\overline{DR} \to 110\ (d)\\ DR \to 111\ (e)\end{array}$ | $(f): \begin{array}{l}\overline{CR} \to 100\ (f)\\ CR \to 101\ (g)\end{array}$ |
| 1 | $(b): \begin{array}{l}CP \to 001\ (b)\\ \overline{CP} \to 011\ (c)\end{array}$ | $(c): \begin{array}{l}GT \to 100\ (f)\\ EQ \to 110\ (d)\\ LT \to 000\ (a)\end{array}$ | $(e): \to 000\ (a)$ | $(g): \to 011\ (c)$ |

| $B^+ D_B$ | AB | | | |
|---|---|---|---|---|
| C | 00 | 01 | 11 | 10 |
| 0 | 0 | Ø | 1 | 0 |
| 1 | $\overline{\text{CP}}$ | EQ | 0 | 1 |

$EQ$ : EQ_1_EURO
$CP$ : COIN_PRESENT

$$B^+ = D_B = \overline{A}\cdot B\cdot EQ + B\cdot\overline{C} + \overline{B}\cdot C\cdot\overline{CP} + A\cdot\overline{B}\cdot C$$

*Easy to miss!*

# Next state - $D_C$

$$A^+B^+C^+ = f(ABC, CP, DR, CR, GT, LT, EQ)$$

| $C$ \ $AB$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $(a)$: $\overline{CP}\rightarrow 000\,(a)$; $CP\rightarrow 001\,(b)$ | $\Phi : \Phi$ | $(d)$: $\overline{DR}\rightarrow 110\,(d)$; $DR\rightarrow 111\,(e)$ | $(f)$: $\overline{CR}\rightarrow 100\,(f)$; $CR\rightarrow 101\,(g)$ |
| 1 | $(b)$: $CP\rightarrow 001\,(b)$; $\overline{CP}\rightarrow 011\,(c)$ | $(c)$: $GT\rightarrow 100\,(f)$; $EQ\rightarrow 110\,(d)$; $LT\rightarrow 000\,(a)$ | $(e)$: $\rightarrow 000\,(a)$ | $(g)$: $\rightarrow 011\,(c)$ |

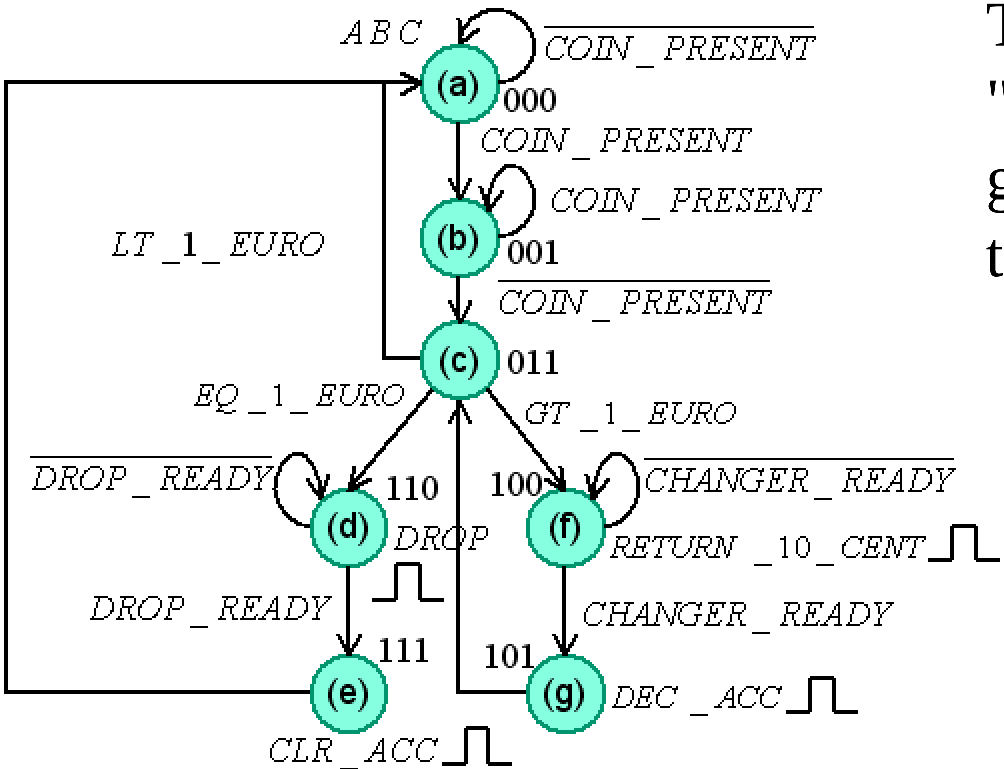
| $C^+$ $D_C$ | AB: 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| C = 0 | CP | Ø | DR | CR |
| C = 1 | 1 | 0 | 0 | 1 |

*CP* : COIN_PRESENT
*DR*: DROP_READY
*CR*: CHANGER_READY

$$C^+ = D_C = \overline{A}\cdot\overline{C}\cdot CP + B\cdot\overline{C}\cdot DR + A\cdot\overline{B}\cdot CR + \overline{B}\cdot C$$

William Sandqvist william@kth.se

# Output signals




The output signals are "pulses" which are generated when passing through the states d, f, e, g.

$$DROP = \boxed{AB\overline{C}}$$

$$CLR\_ACC = \boxed{ABC}$$

$$RETURN\_10\_CENT = \boxed{A\overline{B}\,\overline{C}}$$

$$DEC\_ACC = \boxed{A\overline{B}C}$$

# Implementation of the vending machine

**”students must master the the design of simple combinational and sequential digital systems”**

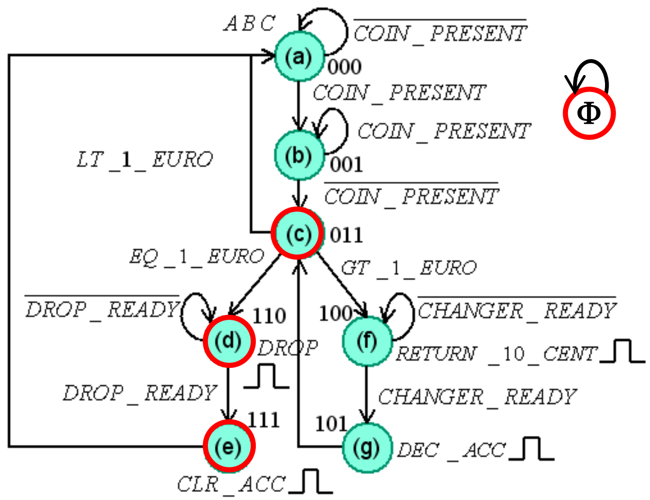
*You do that now!*

***When they call from CocaCola, it's just for you all ”Digital Designers” to adopt the mission ...***

# What will happen in state Φ?

| | | AB | | | |
|---|---|---|---|---|---|
| | | 00 | 01 | 11 | 10 |
| C | 0 | a | Ø | d | f |
| | 1 | b | c | e | g |

$$\Phi = (010)_{ABC}$$




$$A^+ = \overline{A}\cdot B\cdot EQ + \overline{A}\cdot B\cdot GT + A\cdot\overline{C} \quad\Rightarrow\quad A^+(010)_{ABC} = 1\cdot 1\cdot EQ + 1\cdot 1\cdot GT + 0\cdot 1 = EQ + GT$$

$$B^+ = \overline{A}\cdot B\cdot EQ + B\cdot\overline{C} + \overline{B}\cdot C\cdot\overline{CP} + A\cdot\overline{B}\cdot C \quad\Rightarrow\quad B^+(010)_{ABC} = 1\cdot 1\cdot EQ + 1\cdot 1 + \ldots = 1$$

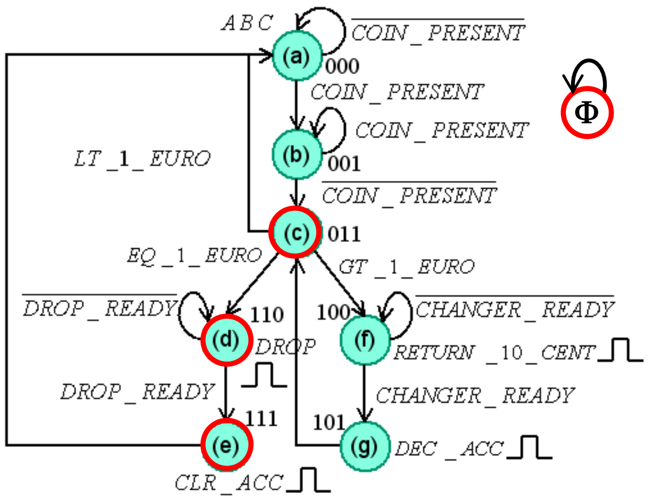
$$C^+ = \overline{A}\cdot\overline{C}\cdot CP + B\cdot\overline{C}\cdot DR + A\cdot\overline{B}\cdot CR + \overline{B}\cdot C$$

$$\Rightarrow\quad C^+(010)_{ABC} = 1\cdot 1\cdot CP + 1\cdot 1\cdot DR + 0\cdot 0\cdot CR + 0\cdot 0 = CP + DR$$

$$A^+B^+C^+ = -1- = 010, 110, 011, 111 \rightarrow \Phi, d, c, e$$

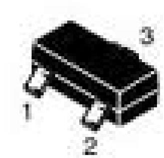
# What will happen in state Φ?

$$A^+B^+C^+ = -1- = 010, 110, 011, 111 \rightarrow \Phi, d, c, e$$




In Φ-state we are stuck, or we go to (c) and then on. Or we go to (d) and offers soft drinks, or we go to (e) and resets any previous payment.

Obviously, we need to purchase a reset circuit that ensures that the machine always starts in (a) **000**! Otherwise we will have legitimate complaints from the customers!




RESET-generator