Electronics Systems (938II)

Lecture 3.3

Semiconductor Memories – EPROM and EEPROM

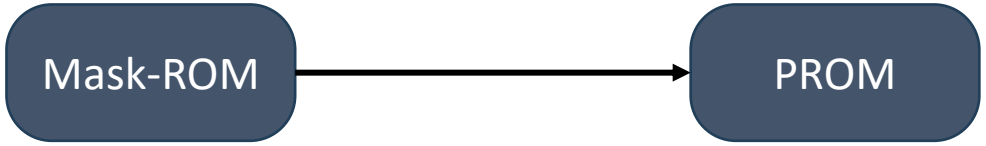
Brief recall

- Evolution of Read-Only Memories (ROMs)

Mask-ROM

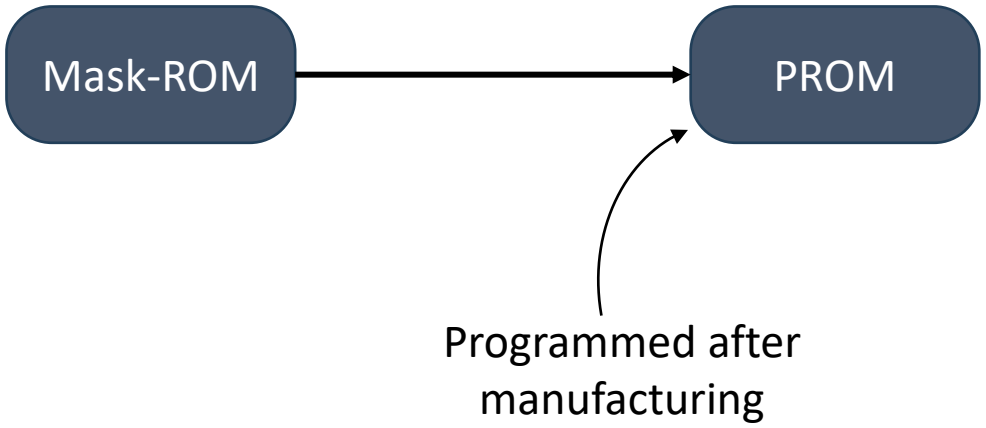
Brief recall

- Evolution of Read-Only Memories (ROMs)



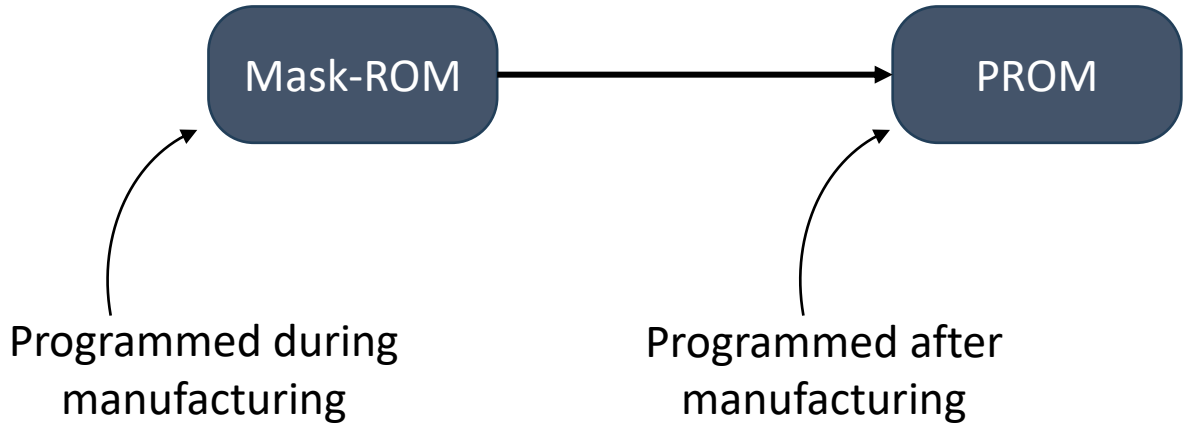
Brief recall

- Evolution of Read-Only Memories (ROMs)



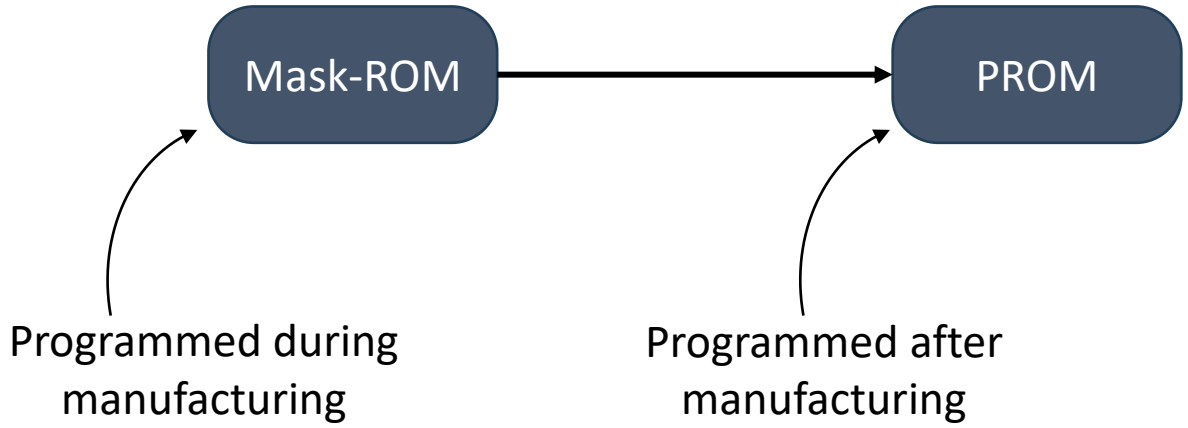
Brief recall

- Evolution of Read-Only Memories (ROMs)



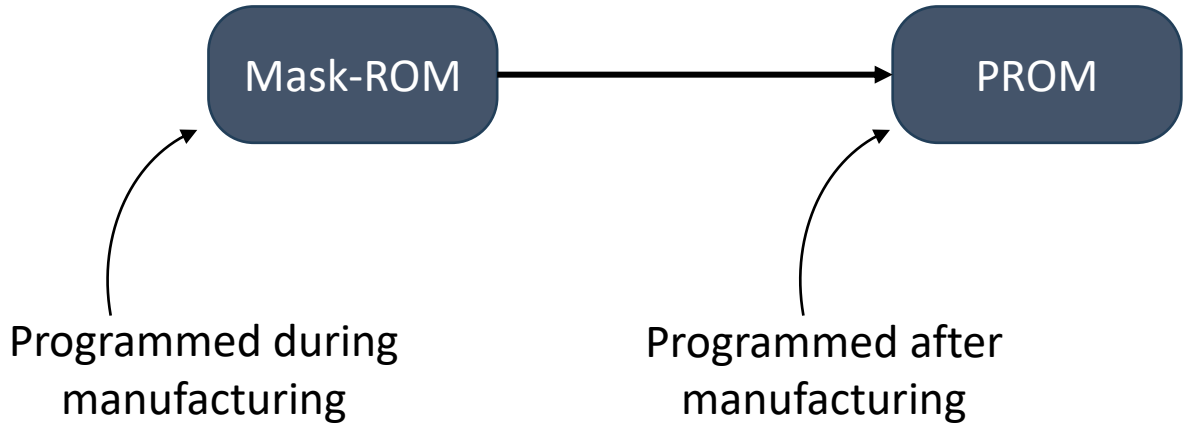
What after?

- Evolution of Read-Only Memories (ROMs)



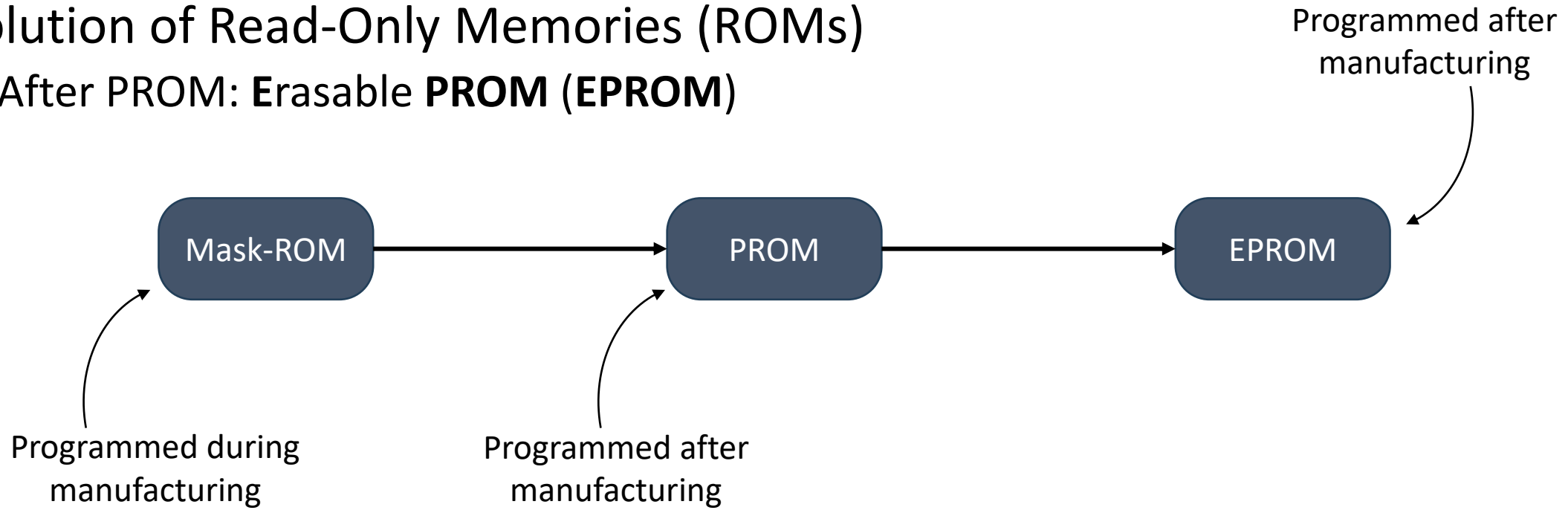
EPROM

- Evolution of Read-Only Memories (ROMs)
  - After PROM: **E**rasable **PROM** (**EPROM**)



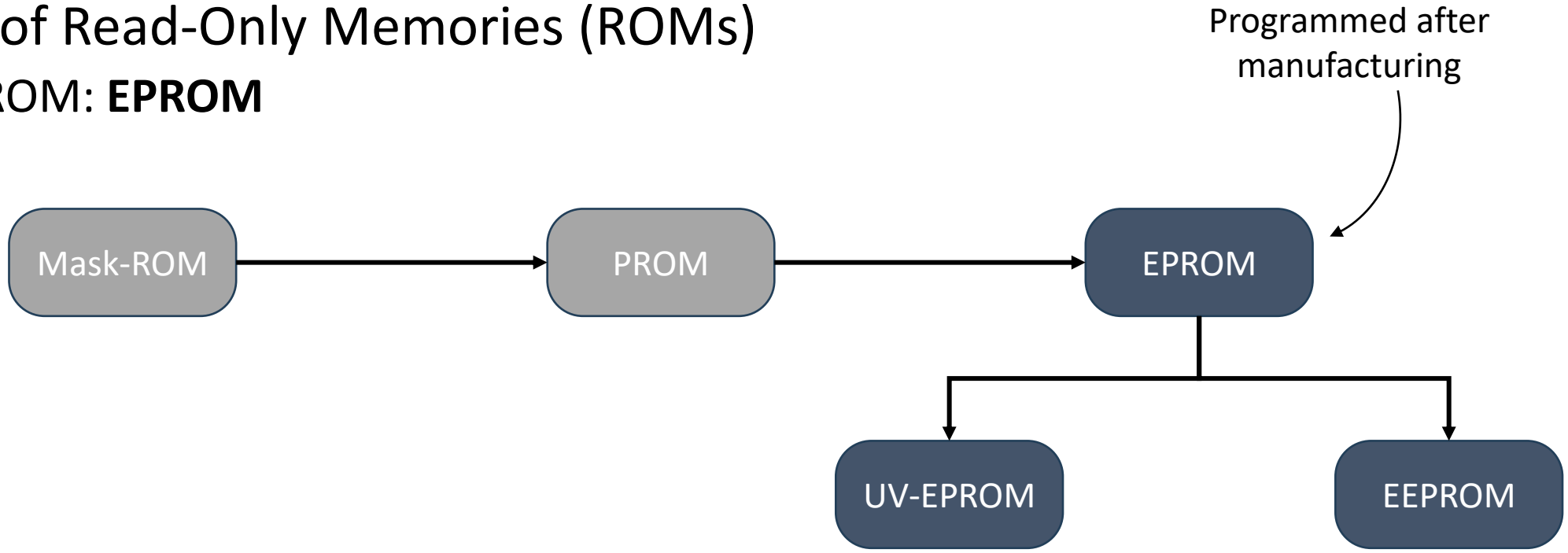
EPROM

- Evolution of Read-Only Memories (ROMs)
  - After PROM: **E**rasable **PROM** (**EPROM**)



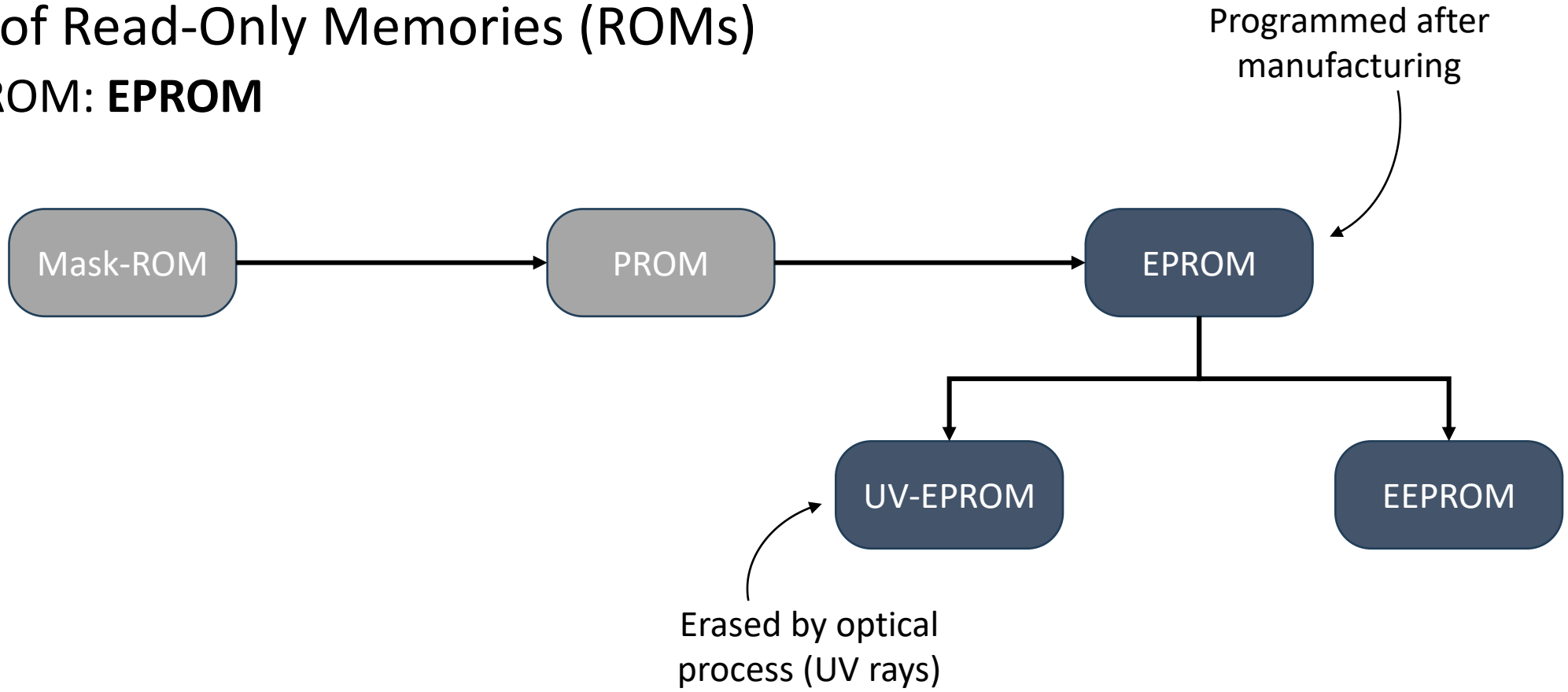
EPROM

- Evolution of Read-Only Memories (ROMs)
  - After PROM: **EPROM**



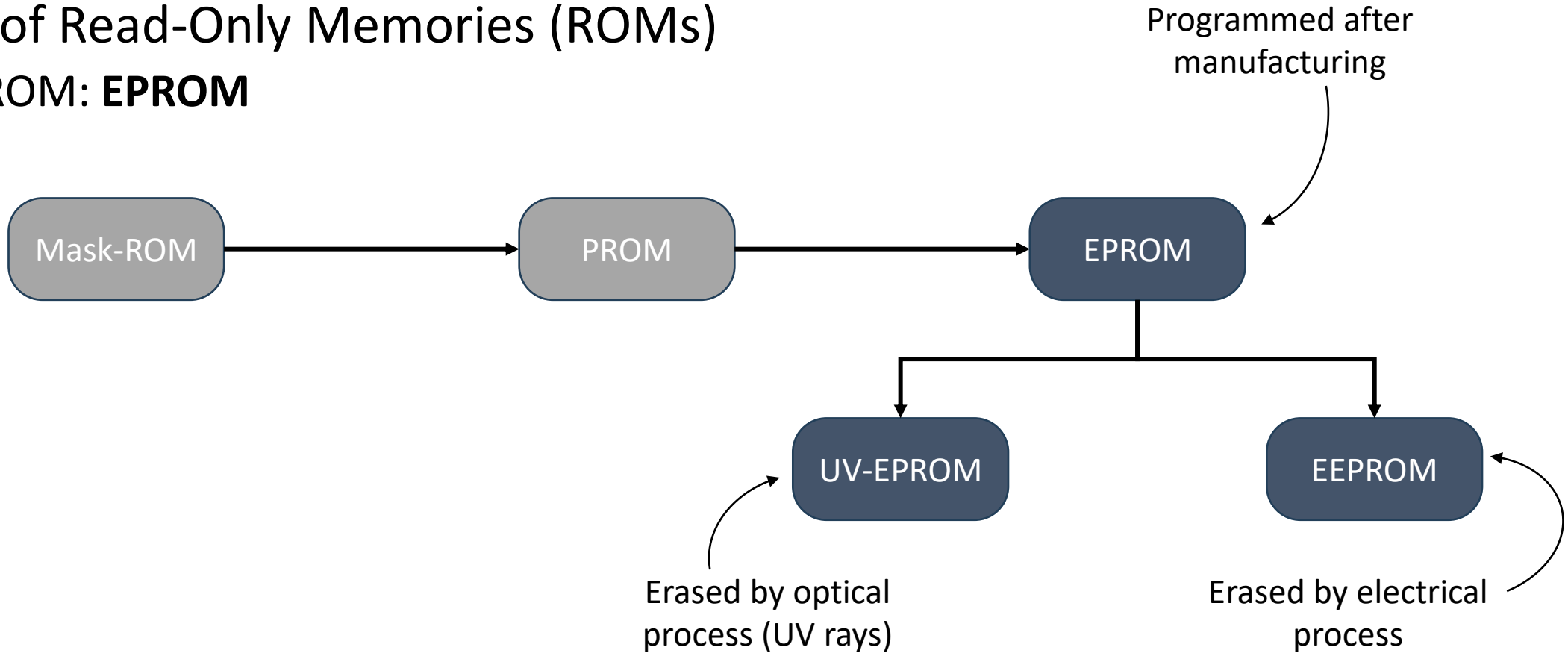
EPROM

- Evolution of Read-Only Memories (ROMs)
  - After PROM: **EPROM**



EPROM

- Evolution of Read-Only Memories (ROMs)
  - After PROM: **EPROM**



EPROM

- Generally speaking
  - It is a ROM
    - Run-time working principle similar to the that of mask-ROM/PROM
  - That can be programmed
  - And erased
    - So reprogrammed !!!

EPROM – Underlying Technology

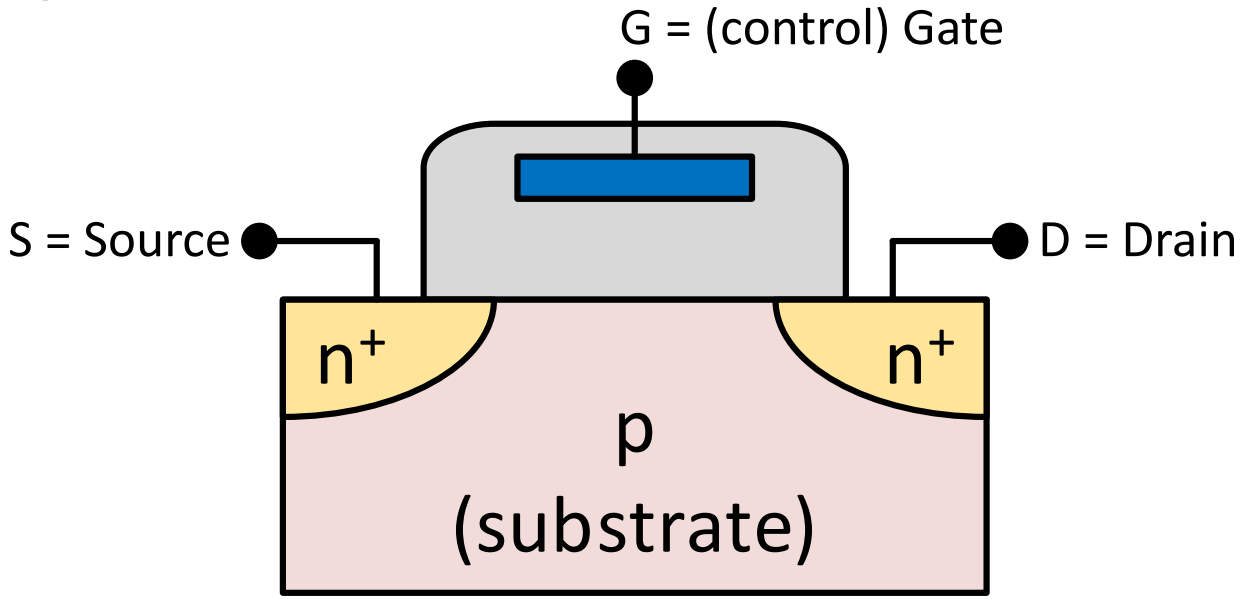
- It relies on particular MOS transistors

EPROM – Underlying Technology

- It relies on particular MOS transistors
  - **FGMOS** = **F**loating **G**ate **MOS**
    - In practice, a MOS + one additional (floating) gate

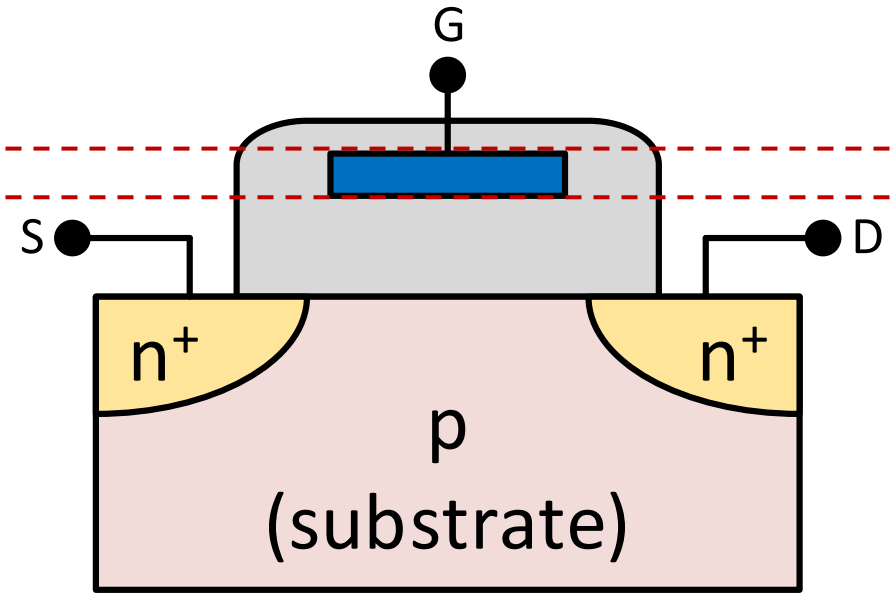
EPROM – Underlying Technology

- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Traditional MOS transistor



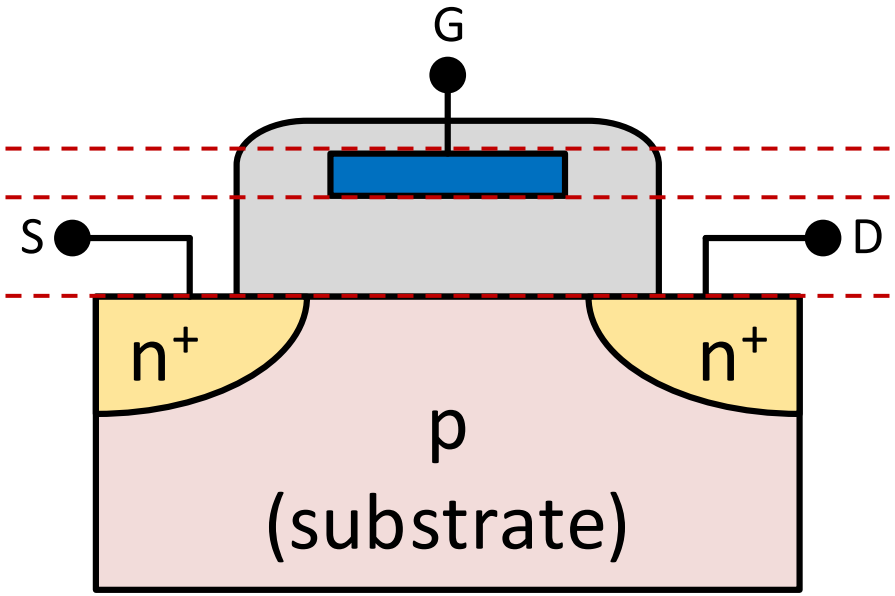
EPROM – Underlying Technology

- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Traditional **M**OS transistor
        - **M**etal



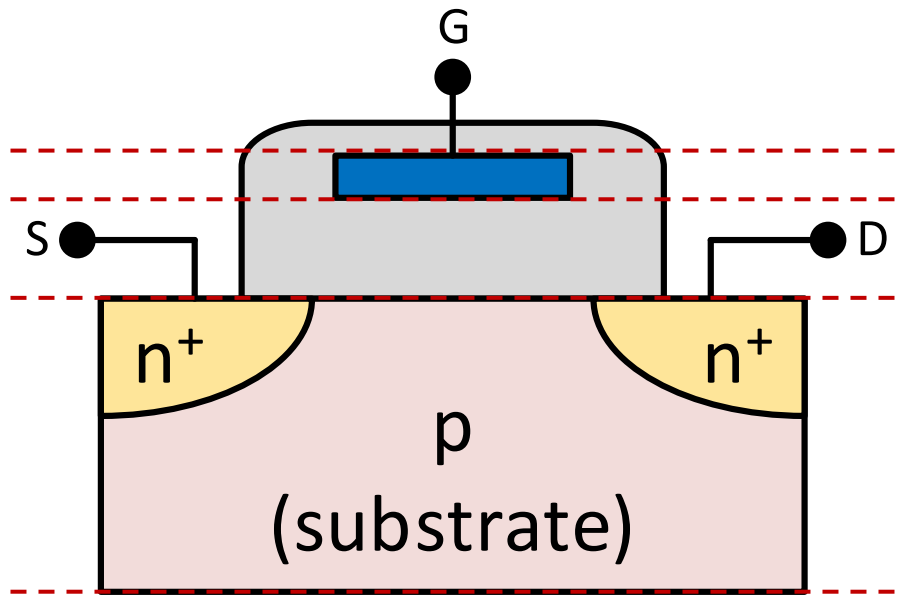
EPROM – Underlying Technology

- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Traditional M**O**S transistor
        - Metal
        - **O**xide



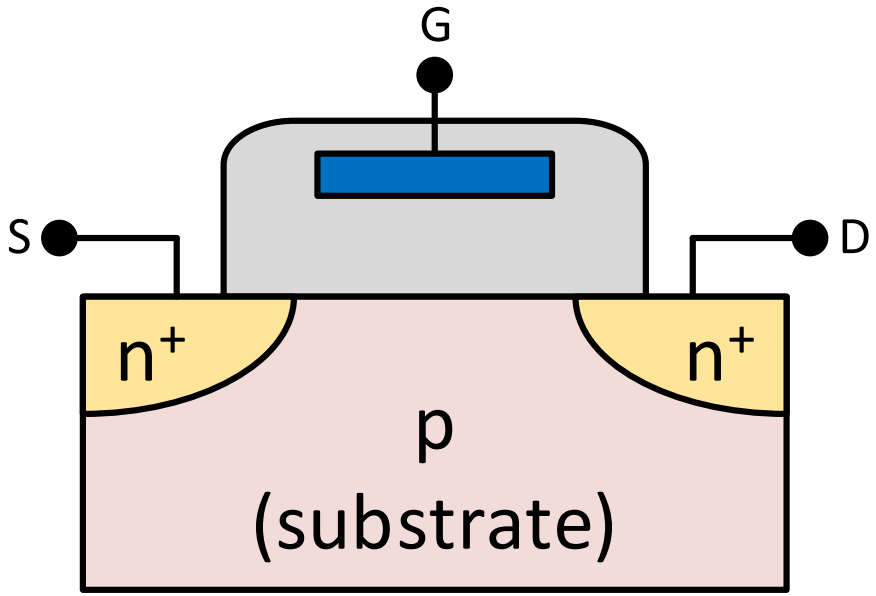
EPROM – Underlying Technology

- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Traditional MO**S** transistor
        - Metal
        - Oxide
        - **S**ilicon



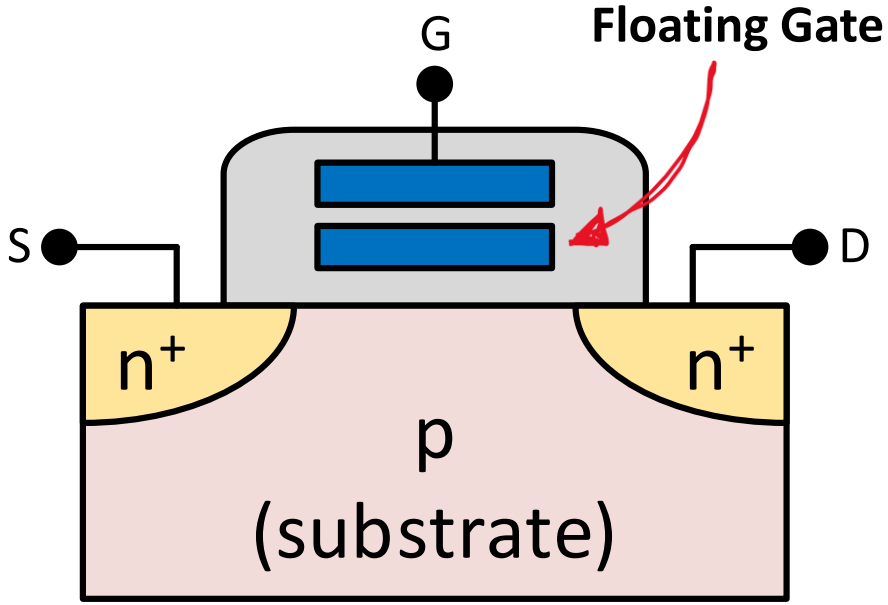
EPROM – Underlying Technology

- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Traditional MOS transistor
        - Metal
        - Oxide
        - Silicon



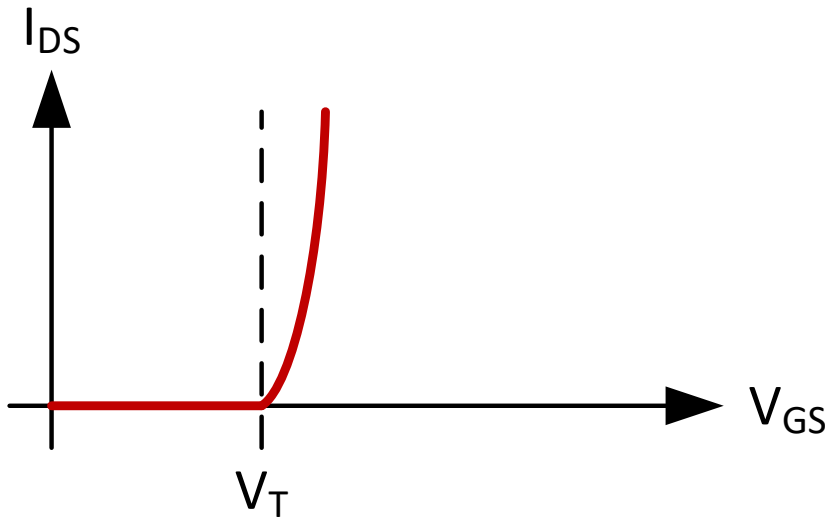
EPROM – Underlying Technology

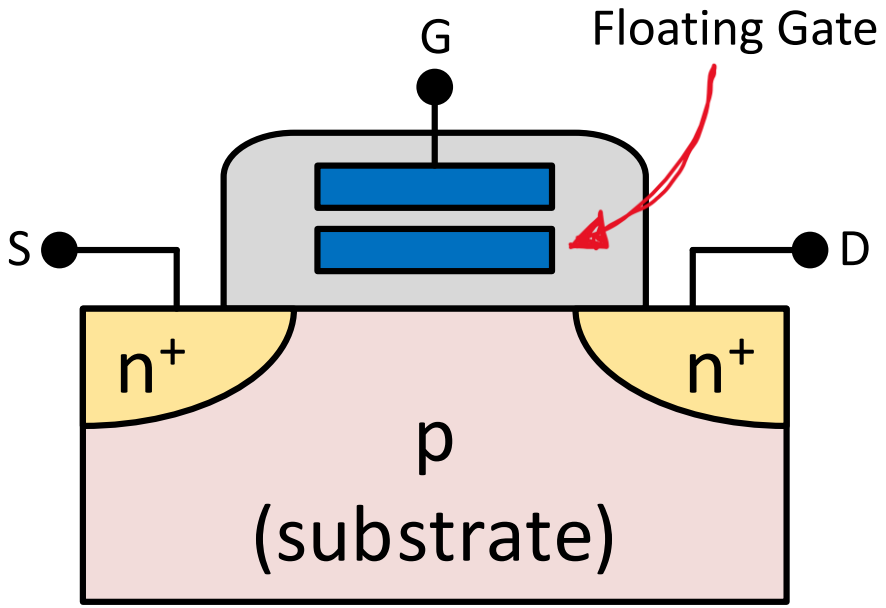
- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate



EPROM – Underlying Technology

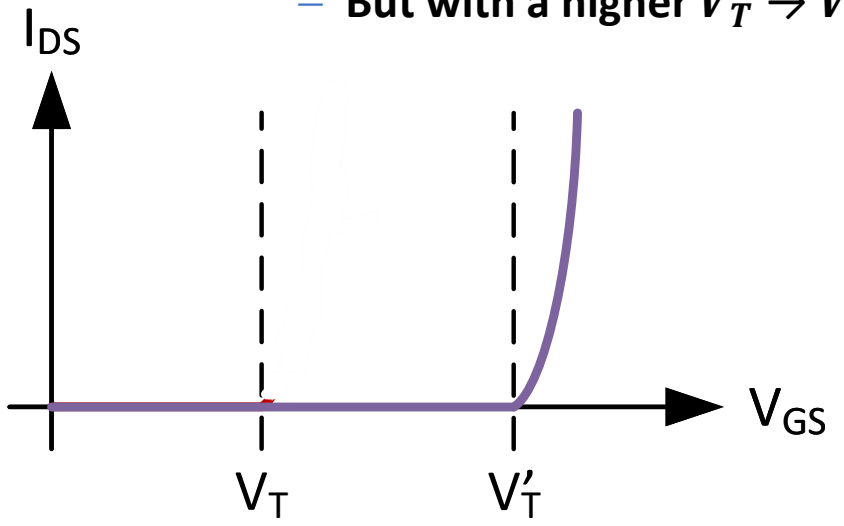
- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Floating gate = **empty** (**default**)
        - It behaves like a "traditional" MOS

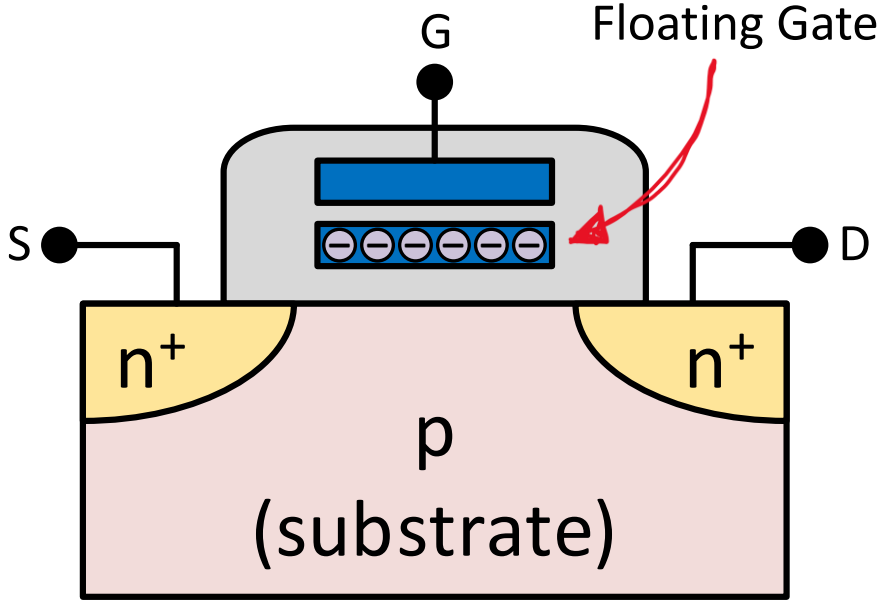




EPROM – Underlying Technology

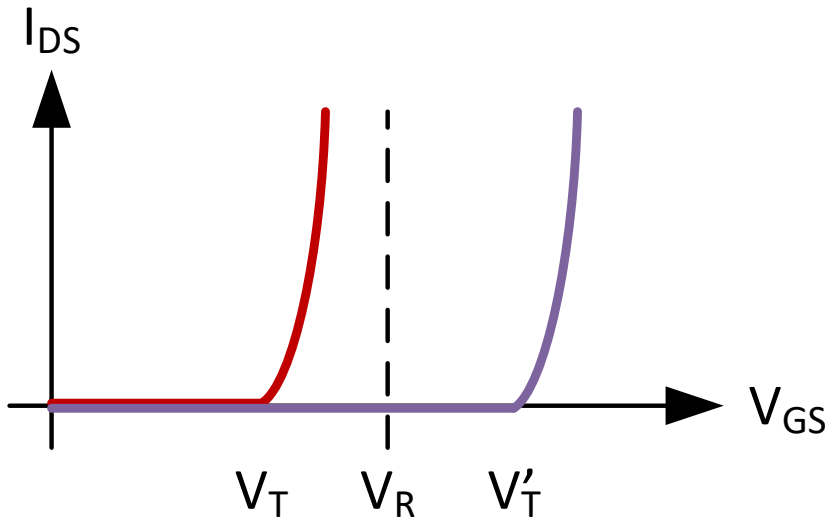
- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + one additional (floating) gate
      - Floating gate = **full of electrons** (**programmed**)
        - It behaves like a "traditional" MOS
        - **But with a higher $V_T \rightarrow V'_T$**

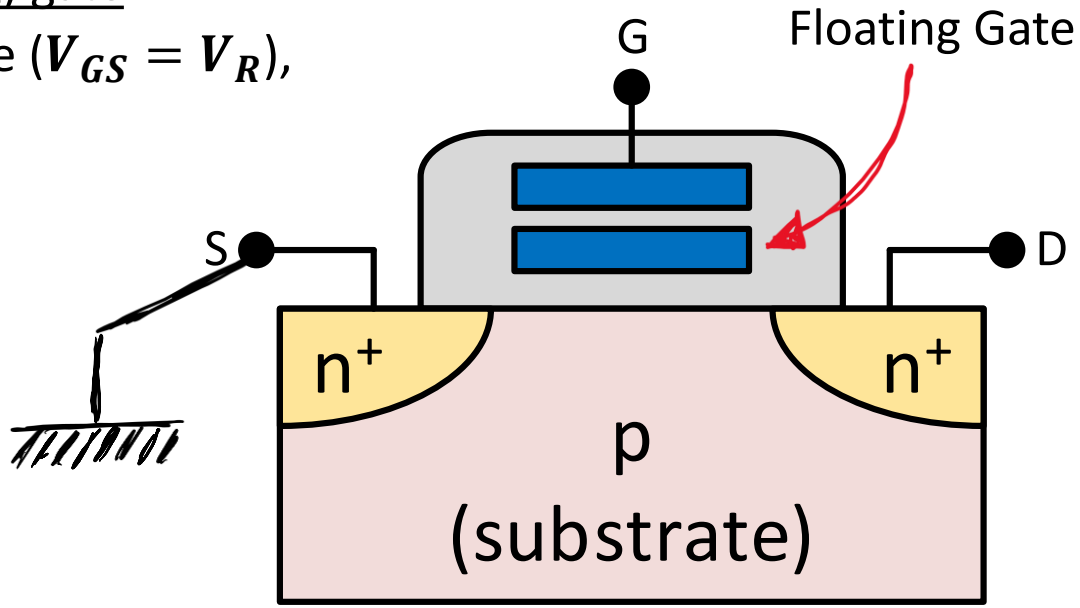




EPROM – Underlying Technology

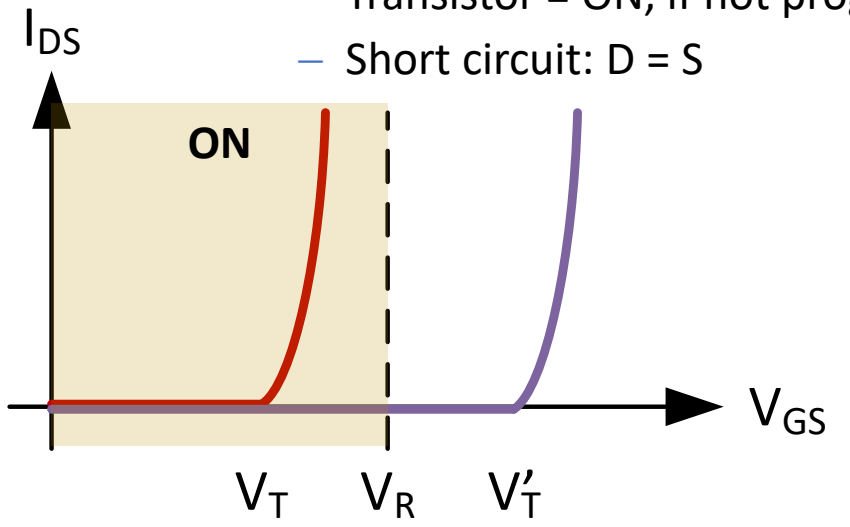
- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + <u>one additional (floating) gate</u>
      - Using a reading voltage (V_R) on the gate ($V_{GS} = V_R$), such that $V_T < V_R < V'_T$

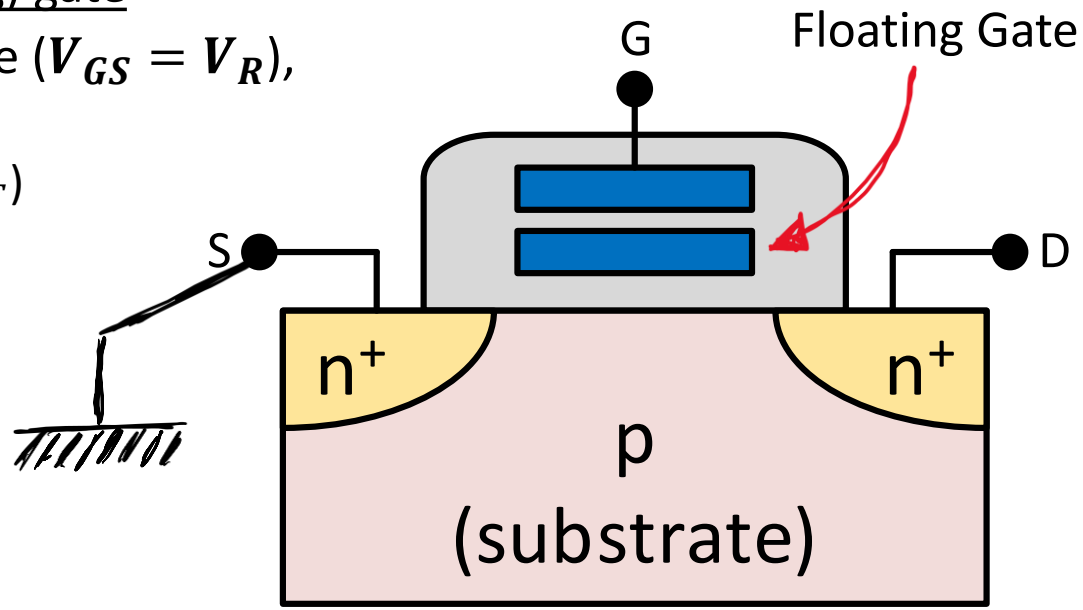




EPROM – Underlying Technology

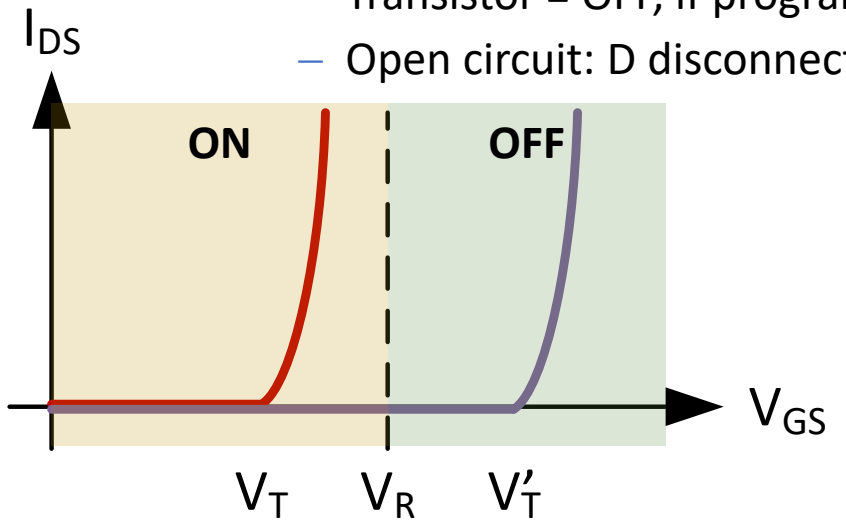
- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + <u>one additional (floating) gate</u>
      - Using a reading voltage ($\boldsymbol{V_R}$) on the gate ($\boldsymbol{V_{GS} = V_R}$), such that $\boldsymbol{V_T < V_R < V_T'}$
        - Transistor = ON, if not programmed (V_T)
        - Short circuit: D = S

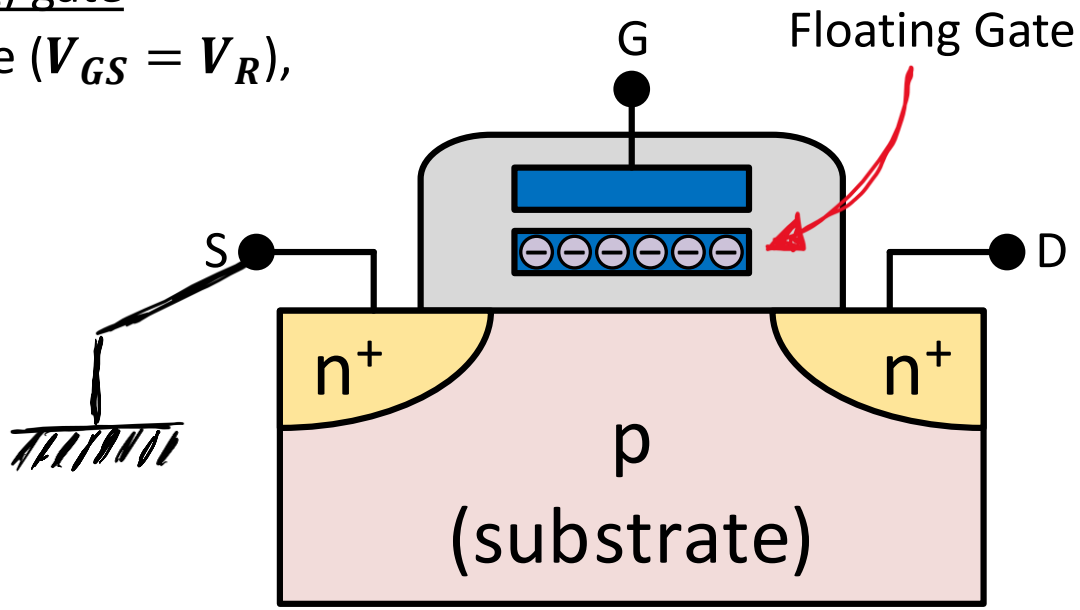




EPROM – Underlying Technology

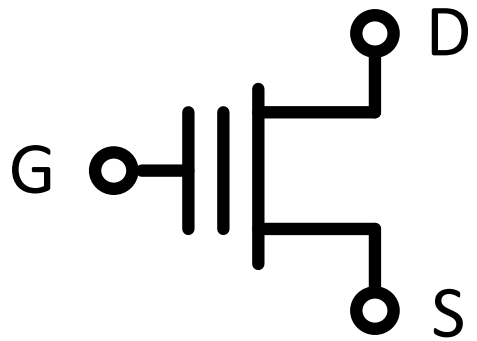
- It relies on particular MOS transistors
  - **FGMOS**
    - In practice, a MOS + <u>one additional (floating) gate</u>
      - Using a reading voltage ($\boldsymbol{V_R}$) on the gate ($\boldsymbol{V_{GS} = V_R}$), such that $\boldsymbol{V_T < V_R < V_T'}$
        - Transistor = OFF, if programmed (V_T')
        - Open circuit: D disconnected from S





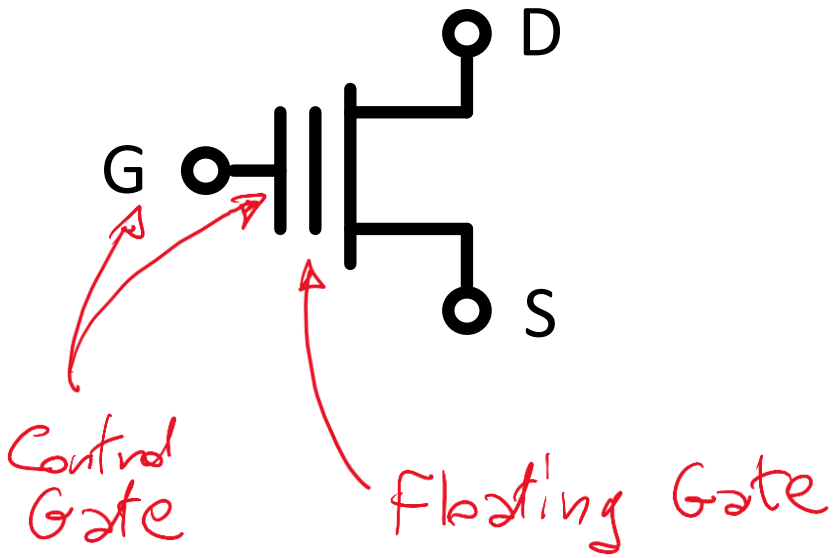
EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - FGMOS symbol



EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - FGMOS symbol

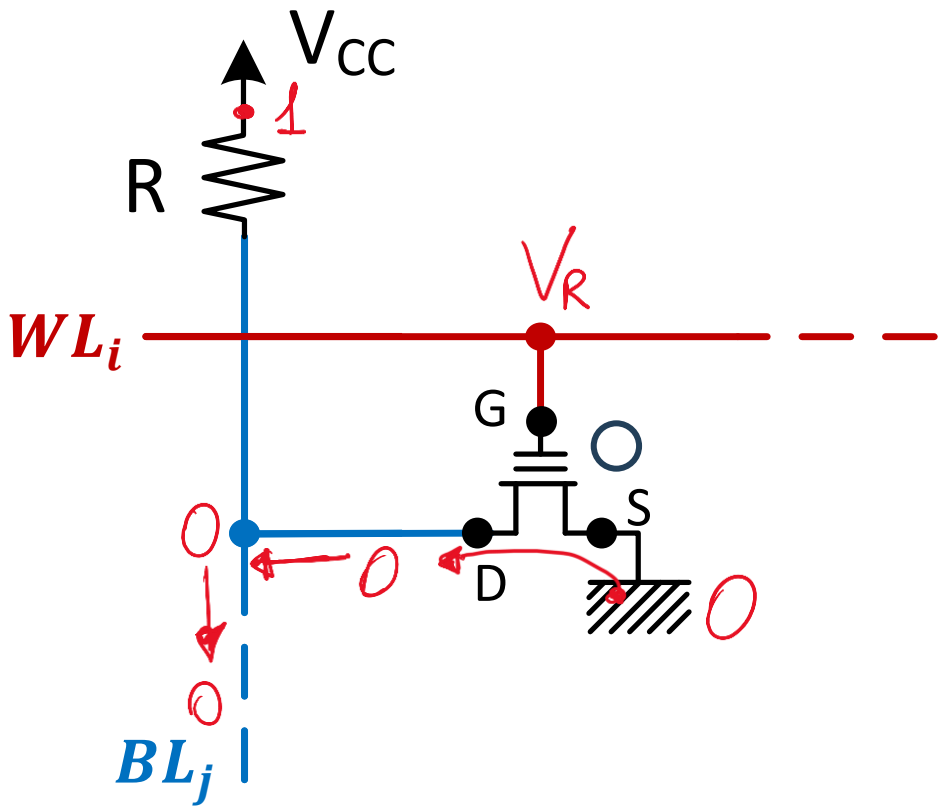


EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - FGMOS symbol

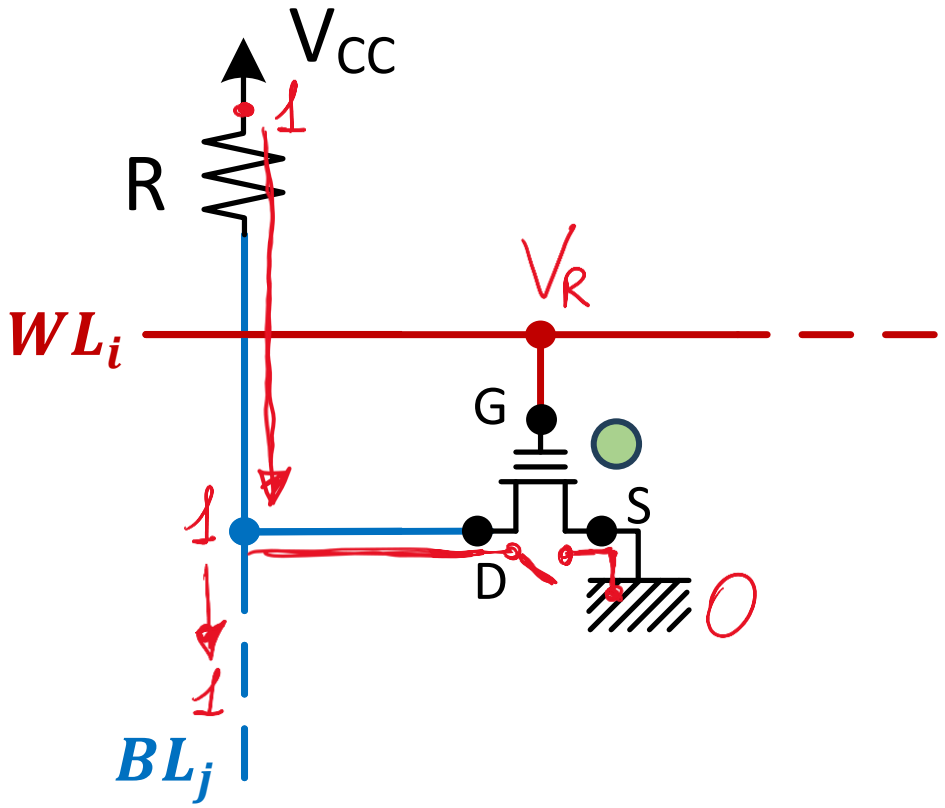
EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - Working principle
    - **FGMOS not programmed** ○
    - Reading with $V_R > V_T$ on WL_i
      - FGMOS = ON
      - FGMOS = short circuit → $D = S \rightarrow V_D = V_S$
      - $\boldsymbol{BL_j = 0}$ **(logic)** $= V_D$



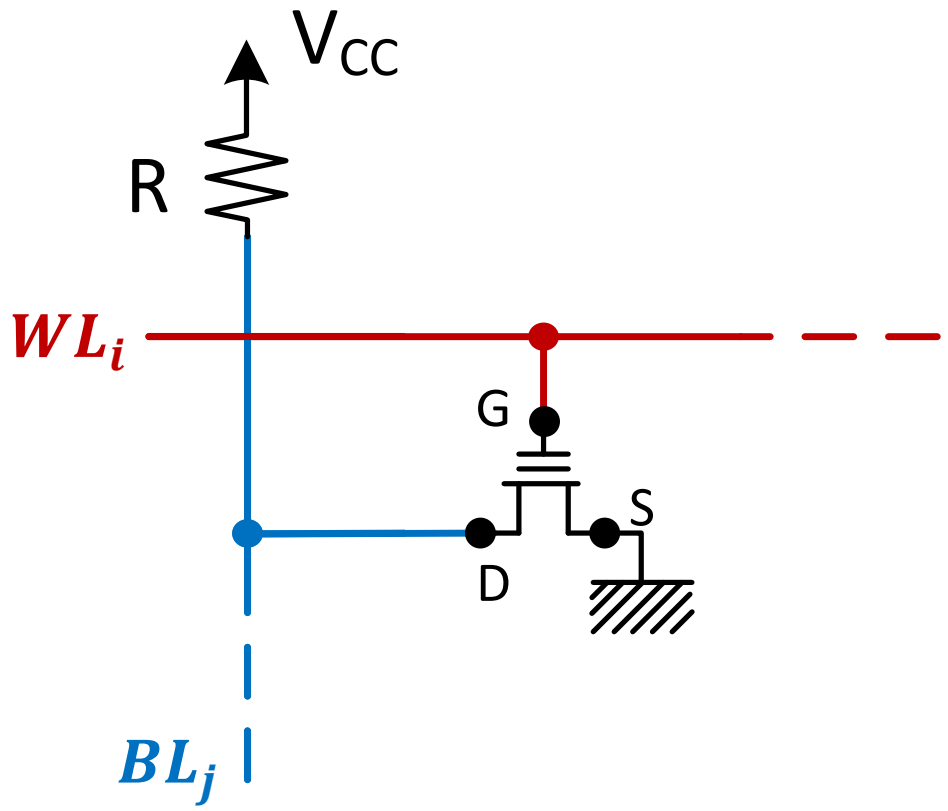
EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - Working principle
    - **FGMOS programmed**
    - Reading with $V_R < V_T'$ on WL_i
      - FGMOS = OFF
      - FGMOS = open circuit → $D \neq S \rightarrow V_D \neq V_S$
      - $\boldsymbol{BL_j = 1}$ **(logic)** from pull-up resistor (R)



EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - Why does this work?
  - Because of …
    1. Time required for programming/erasing
    2. Data retention
    3. Endurance



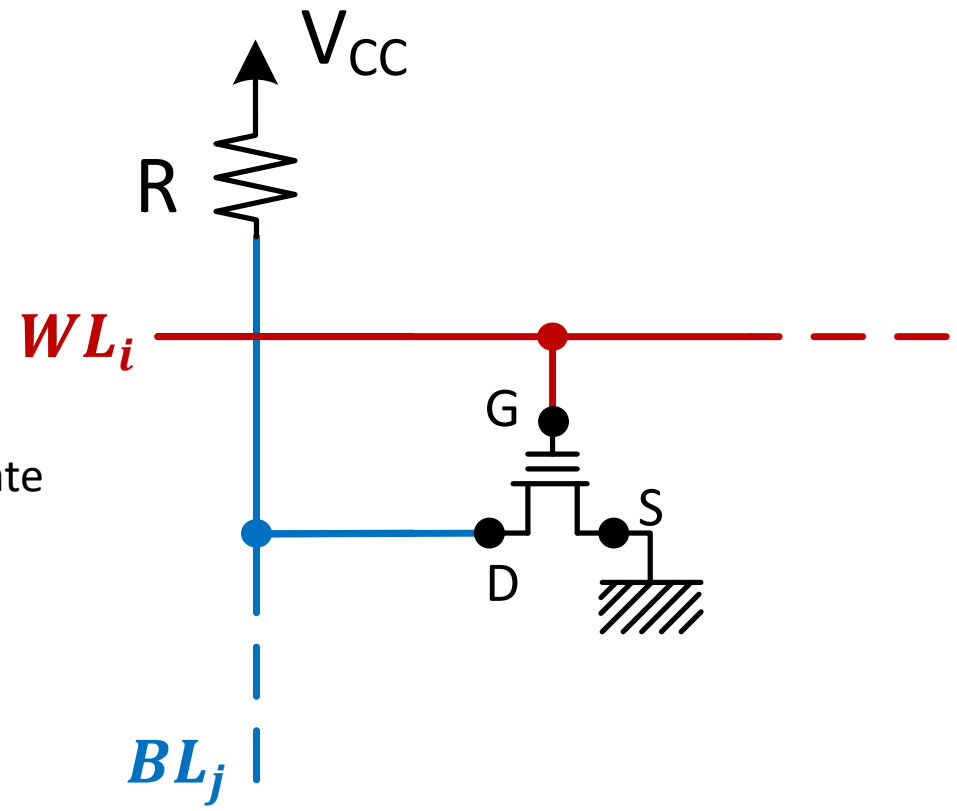
EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - Why does this work?
  - Because of …

1. **Time required for programming/erasing**

   – Programming = injection of electrons into the floating gate
   – Erasing = removal of electrons from the floating gate

   – **Erasing + programming (cycle) takes a time in the order of minutes**



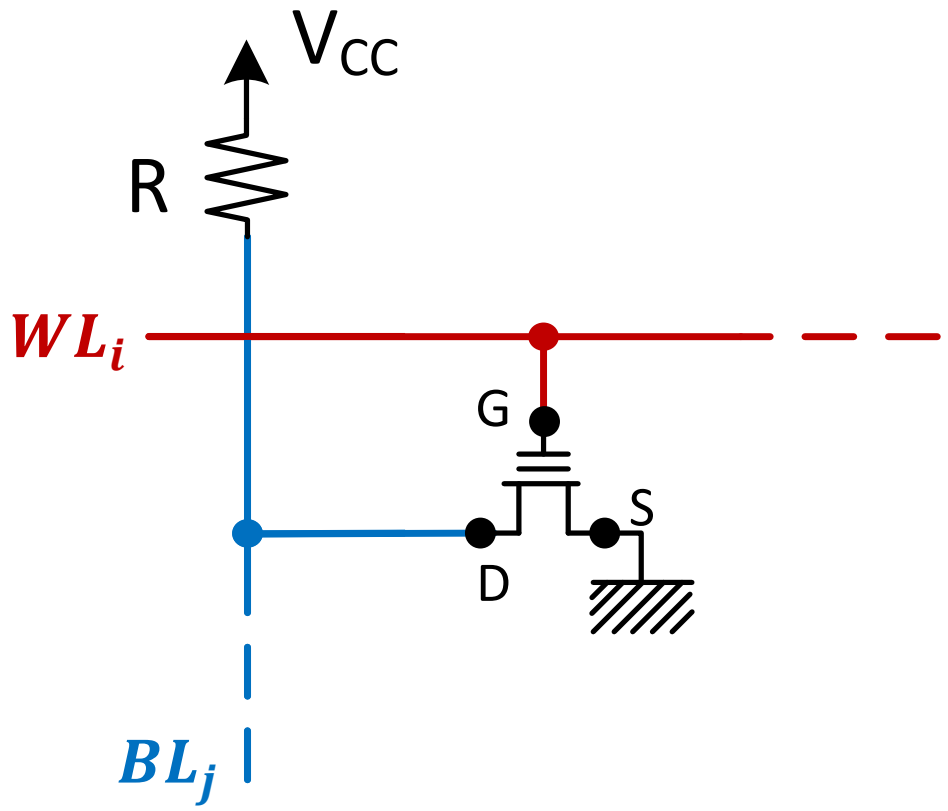
EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - Why does this work?
  - Because of …

    2. **Data retention**

       – Permanence time of electrons in the floating gate (once injected) ≈ 10 ÷ 20 years

       – **Greater than duration of typical electronic application**

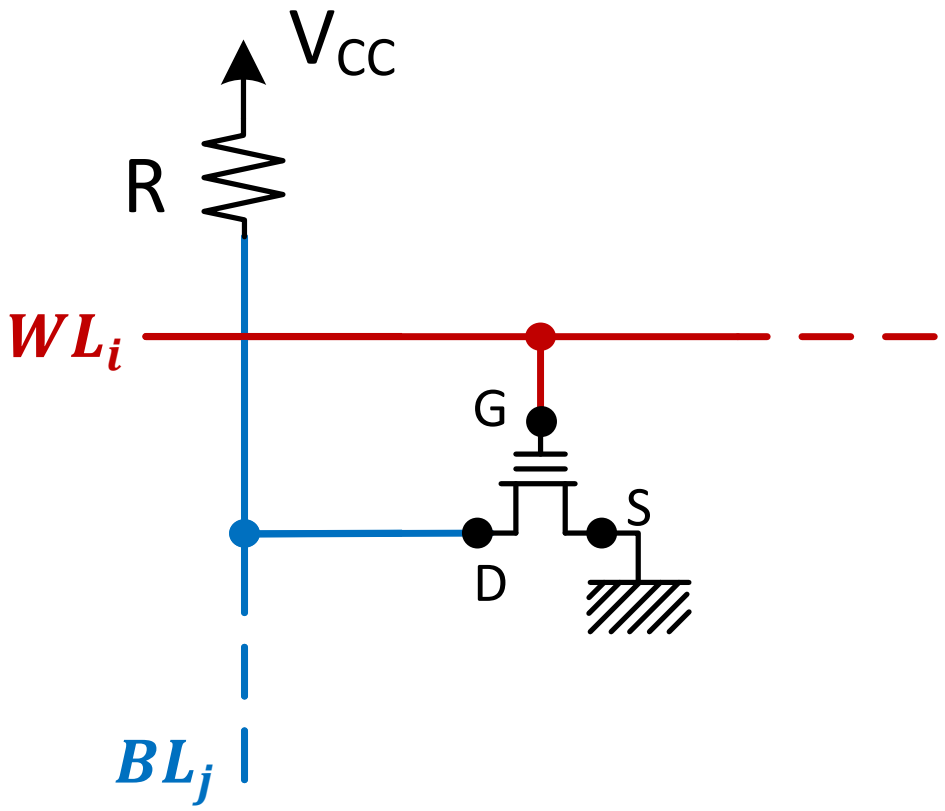


EPROM – Underlying Technology

- Applying FGMOS to semiconductor memories architecture …
  - Why does this work?
  - Because of …

3. **Endurance**

   - Number of cycles (erasing + programming) = endurance
   - Endurance is limited
     - Moving electrons through the oxide degrades it
   - **However, it is not low: typically, in the range $10^3 \div 10^5$**

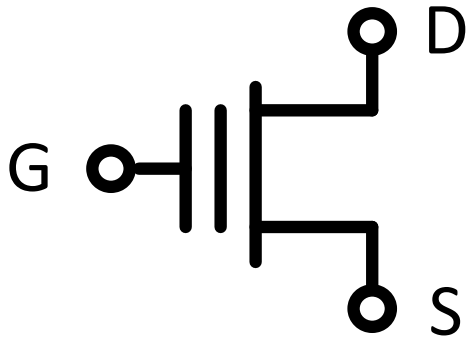


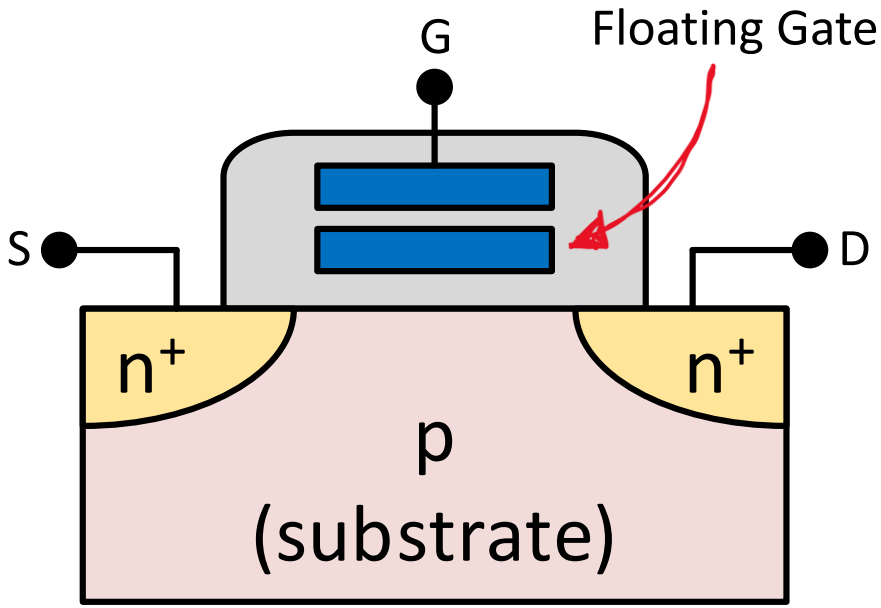
EPROM – Underlying Technology

- Two different kinds of FGMOS
  - Changing the way electrons are injected into (programming) and removed from (erasing) floating gate
    1. **FAMOS** = **F**loating-gate **A**valanche-injection **MOS**
    2. **FLOTOX** = **Flo**ating-gate **T**unnel-**Ox**ide MOS

FAMOS transistor

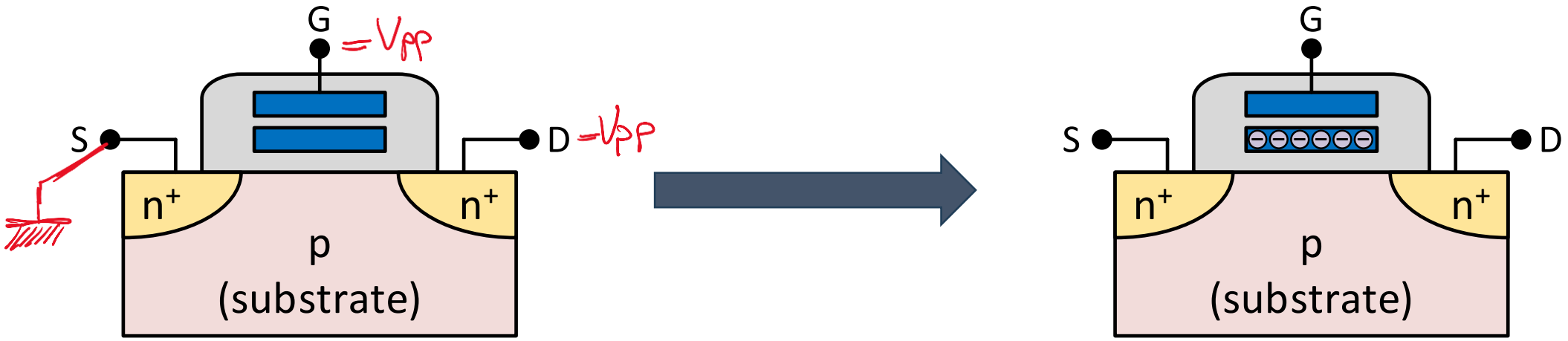
- Same layout and symbol as FGMOS
  - It was the first FGMOS





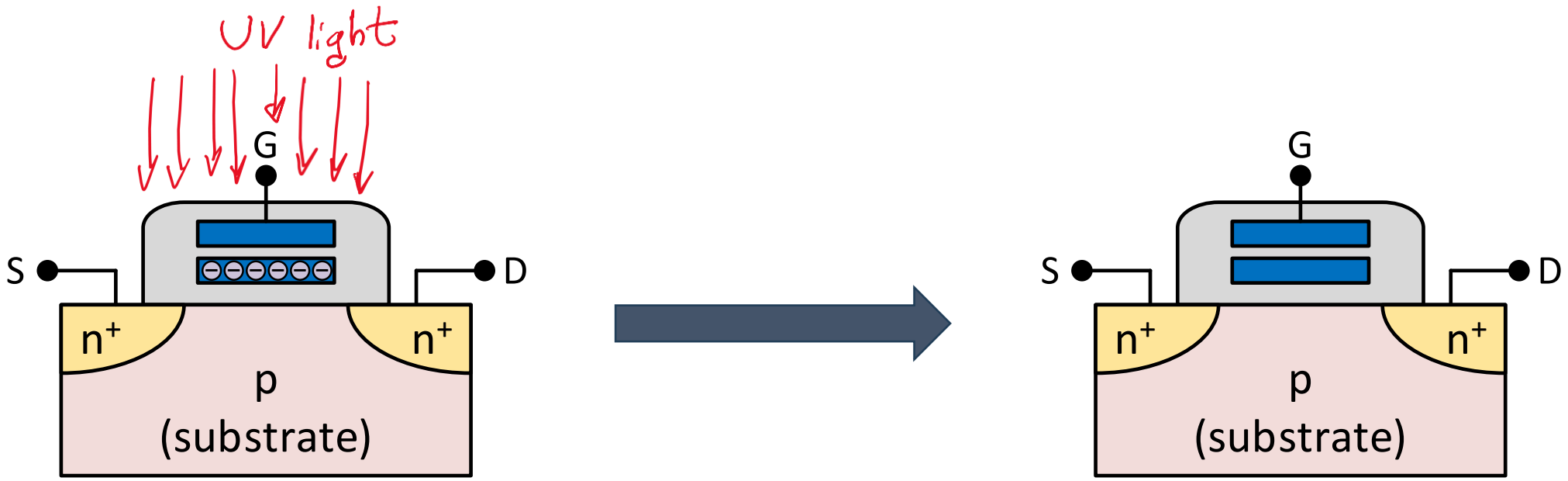
FAMOS transistor

- Working principle
  - Electrons injection (programming)
    - Avalanche effect = very high current → Floating-gate Avalanche-injection MOS
      - $V_G = V_D = V_{PP}$ (programming voltage, $V_{PP} \gg V_R$)
      - $V_S = 0$



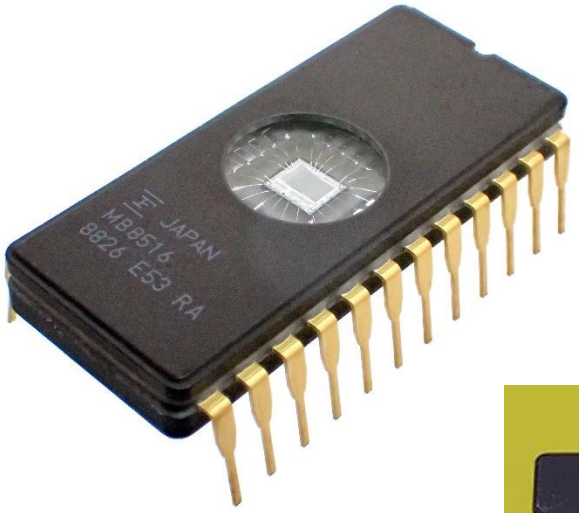
FAMOS transistor

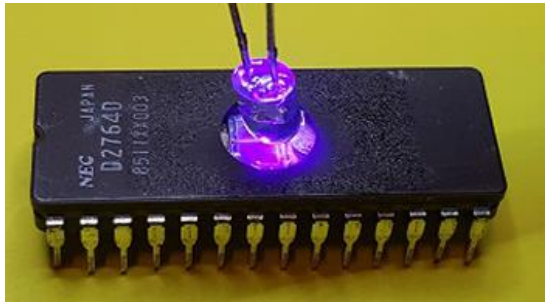
- Working principle
  - Electrons removal (erasing)
    - Exposure to UV light



UV-EPROM (or EPROM)

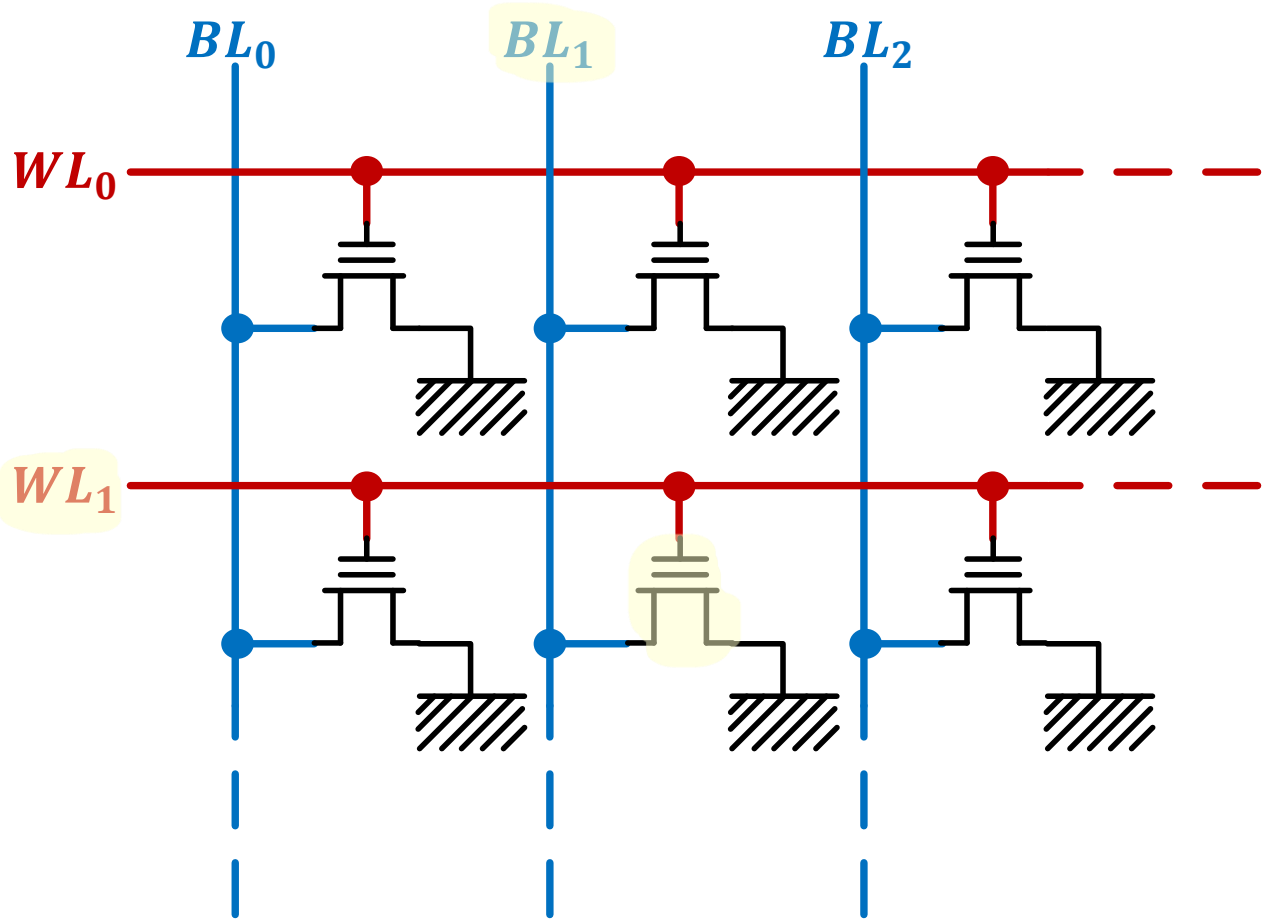
- It is based on FAMOS transistor
  - In practice, it is a PROM with FAMOS transistor instead of MOS + fuse (or anti-fuse)
  - Erasing procedure = exposure to UV light
    - From here the name → **UV-EPROM**
    - The chip is encapsulated in a window
    - Erasing the whole chip (ROM content) at once
  - Also called just **EPROM**
    - It was the first type of EPROM





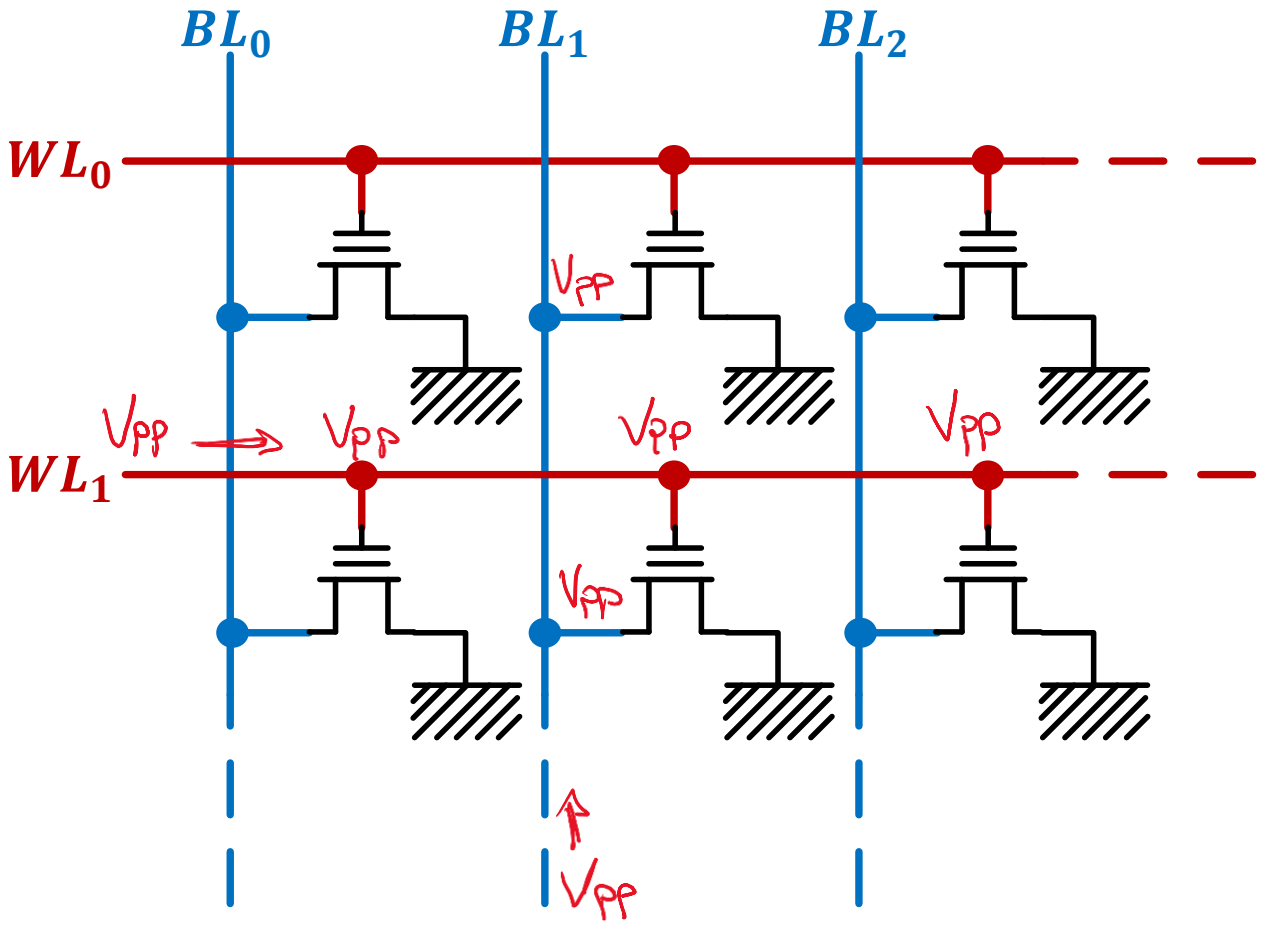
UV-EPROM (or EPROM)

- Working principle
  - Reading: the same as ROM/PROM
    - If transistor programmed → OFF
    - Otherwise (not programmed) → ON
  - Erasing: exposure to UV light
  - Programming – Example
    - Programming cell (1,1)



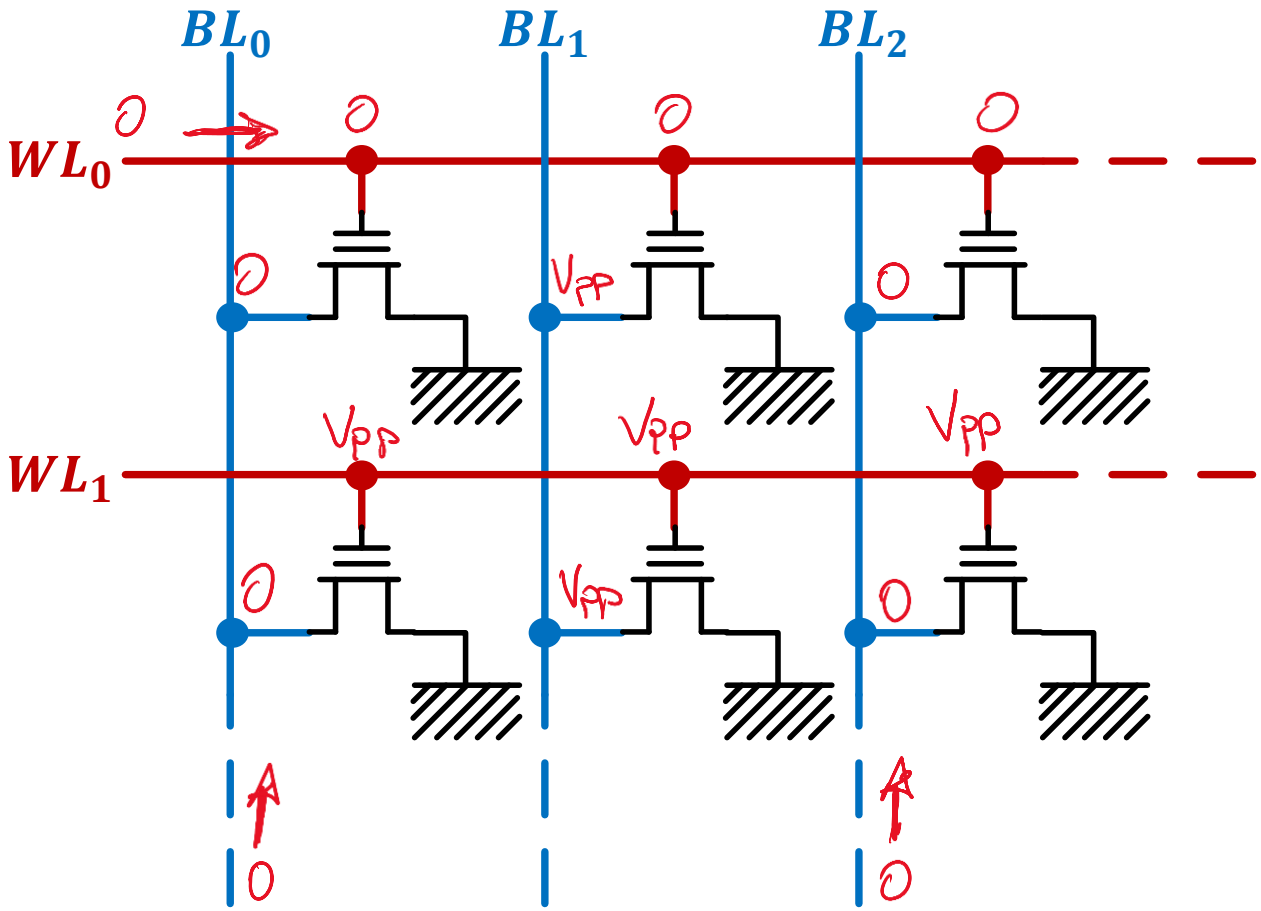
UV-EPROM (or EPROM)

- Working principle
  - Reading: the same as ROM/PROM
    - If transistor programmed → OFF
    - Otherwise (not programmed) → ON
  - Erasing: exposure to UV light
  - Programming – Example
    - Programming cell (1,1)
      - $V_G = WL_1 = V_{PP}$
      - $V_D = BL_1 = V_{PP}$



UV-EPROM (or EPROM)

- Working principle
  - Reading: the same as ROM/PROM
    - If transistor programmed → OFF
    - Otherwise (not programmed) → ON
  - Erasing: exposure to UV light
  - Programming – Example
    - Programming cell (1,1)
      - $V_G = WL_1 = V_{PP}$
      - $V_D = BL_1 = V_{PP}$
      - All other $WL_i = 0$ and $BL_j = 0$!

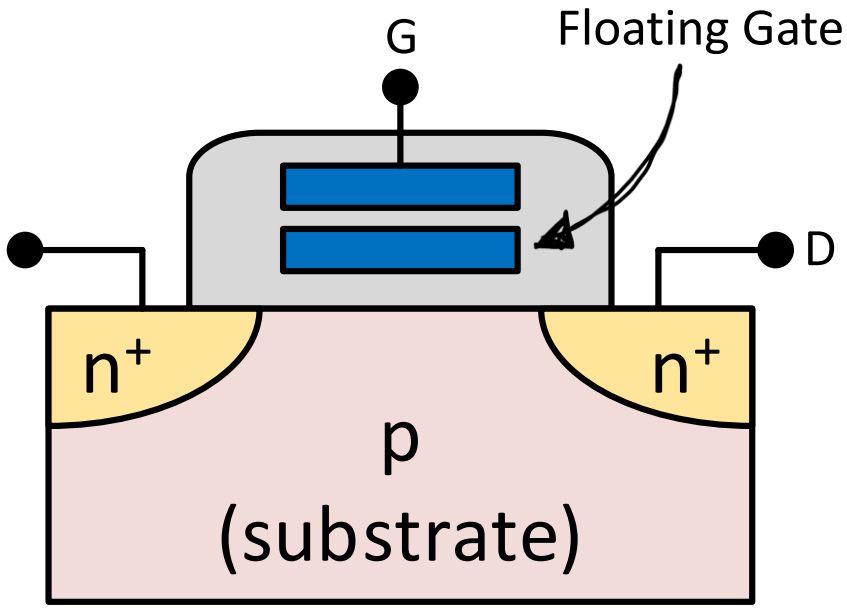


FLOTOX transistor

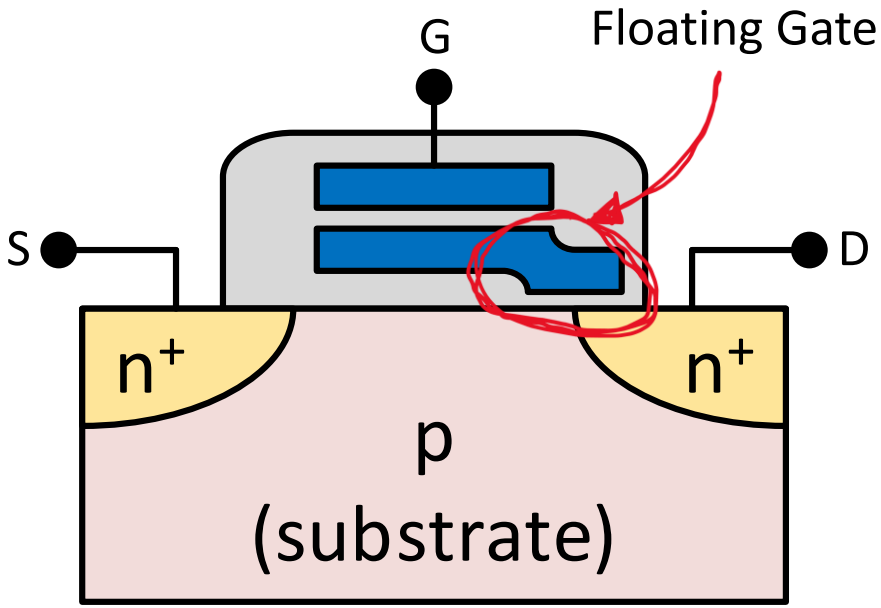
- Some differences w.r.t. FAMOS transistor
  - Floating gate extends over Drain and is separated from it by a very thin oxide layer

FLOTOX transistor

- Some differences w.r.t. FAMOS transistor
  - Floating gate extends over Drain and is separated from it by a very thin oxide layer



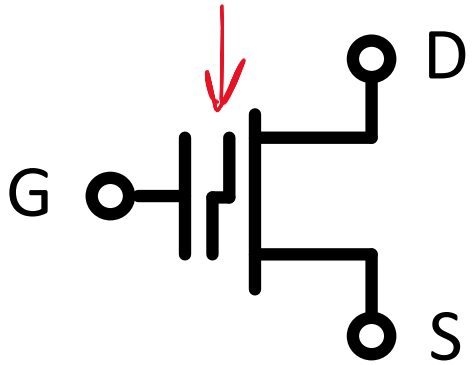
FAMOS

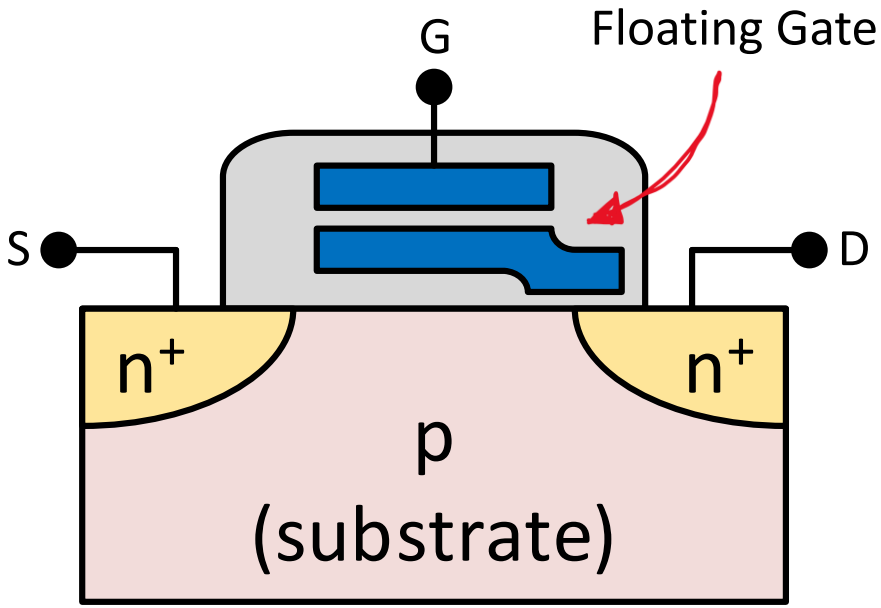


FLOTOX

FLOTOX transistor

- Some differences w.r.t. FAMOS transistor
  - Also the symbol is different



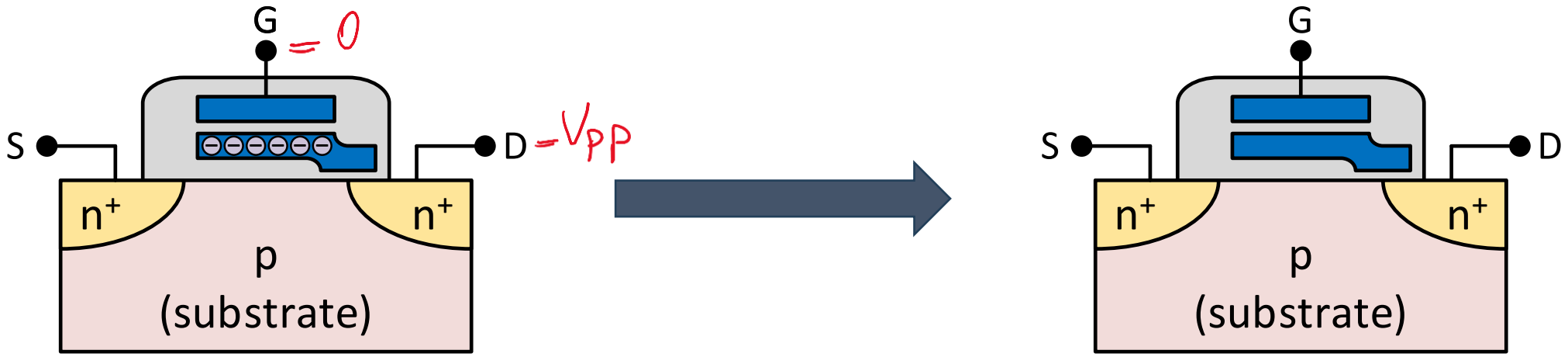


FLOTOX transistor

- Working principle
  - Electrons injection (programming)
    - Tunnel effect (thanks to the very thin oxide layer) → Floating-gate Tunnel-Oxide MOS
      - $V_G = V_{PP}$
      - $V_D = 0$
      - V_S: not specified

FLOTOX transistor

- Working principle
  - Electrons removal (erasing)
    - Tunnel effect (again)
      - $V_G = 0$
      - $V_D = V_{PP}$
      - V_S: not specified



EEPROM (or E^2PROM)

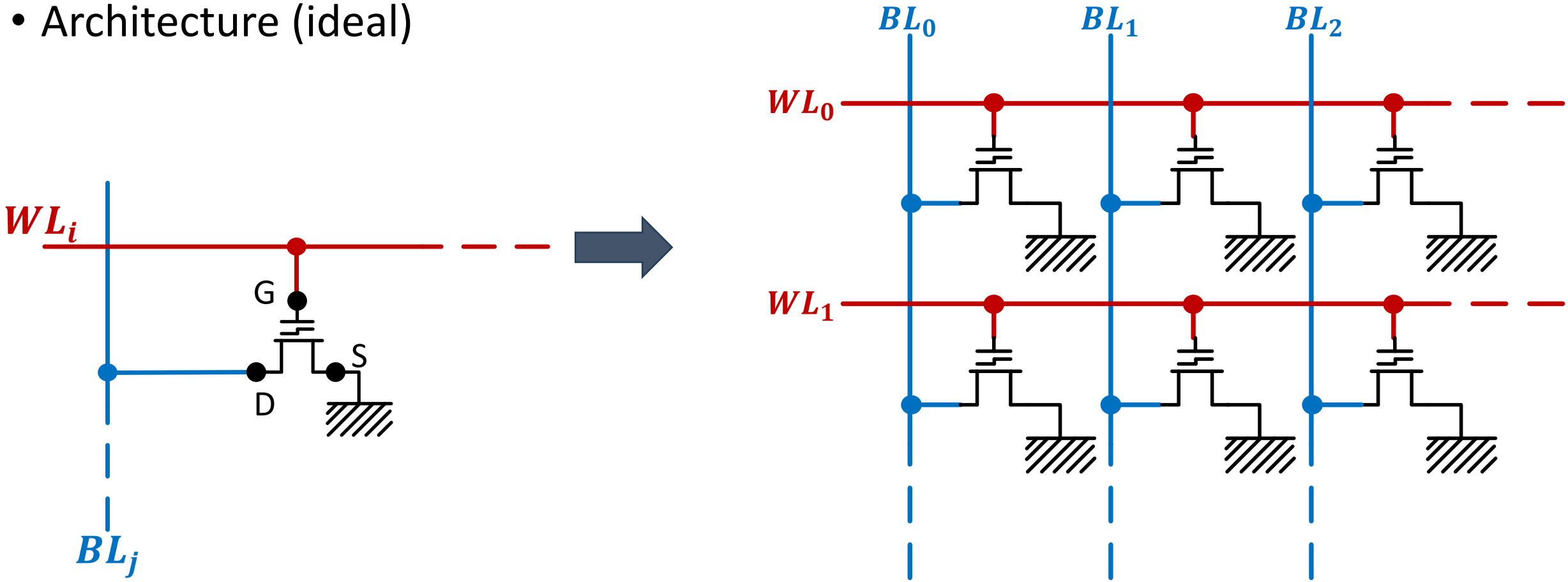
- It is based on FLOTOX transistor
  - In practice, it is a PROM with FLOTOX transistor instead of MOS + fuse (or anti-fuse)
    - Ideally !!!
  - Erasing procedure = tunnel effect due to <u>electrical</u> quantities
    - From here the name → **EEPROM** = **E**lectrically **E**rasable **PROM**
      - Or **E^2PROM**
    - Erasing only selected memory cells (FLOTOX transistors)

EEPROM (or E^2PROM)

- It is based on FLOTOX transistor
  - In practice, it is a PROM with FLOTOX transistor instead of MOS + fuse (or anti-fuse)
    - Ideally !!!
  - Erasing procedure = tunnel effect due to electrical quantities
    - From here the name → **EEPROM** = **E**lectrically **E**rasable **PROM**
      - Or **E^2PROM**
    - Erasing only selected memory cells (FLOTOX transistors)

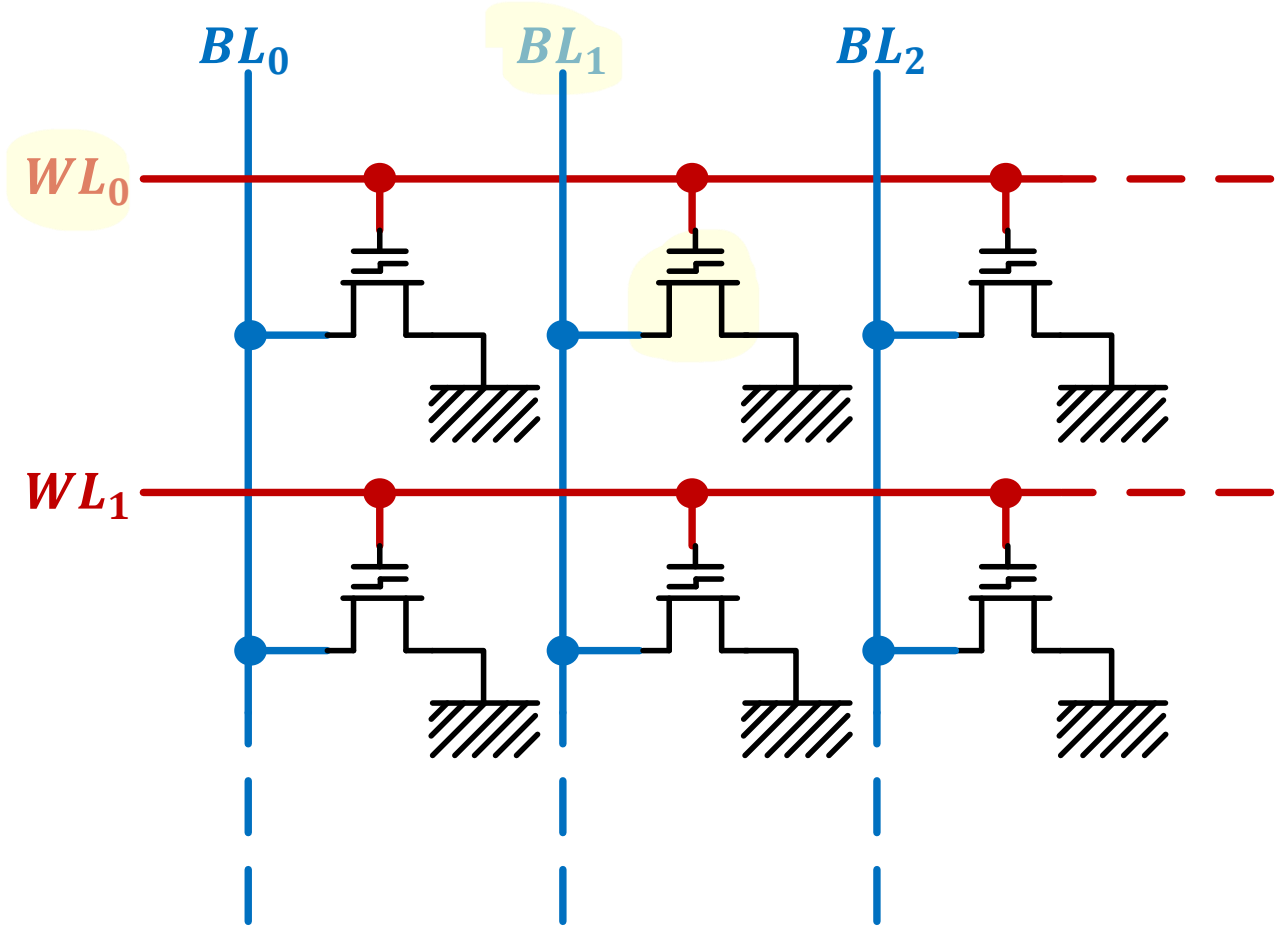
EEPROM (or E²PROM)

- Architecture (ideal)



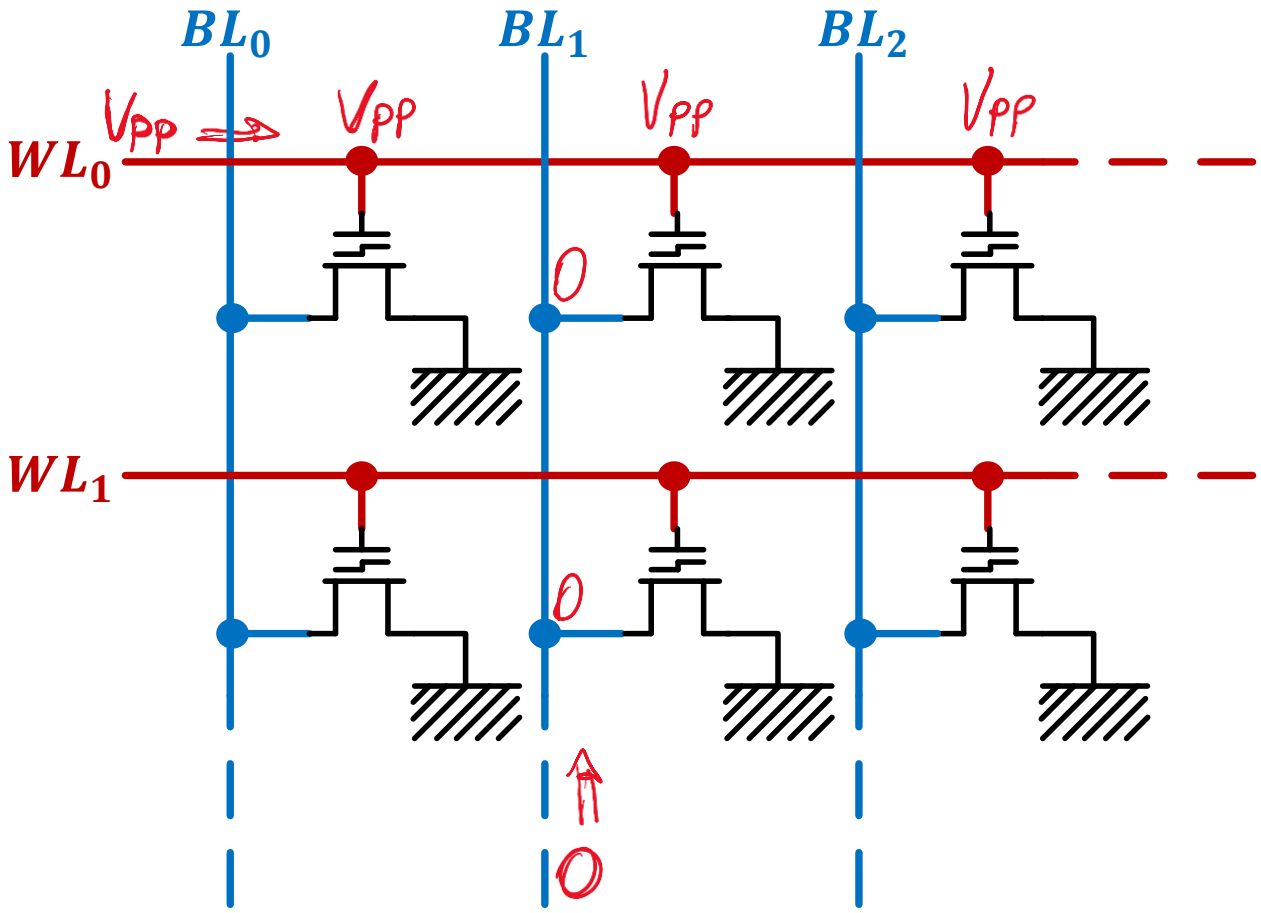
EEPROM (or E²PROM)

- Architecture (ideal)
  - Programming cell (0,1)



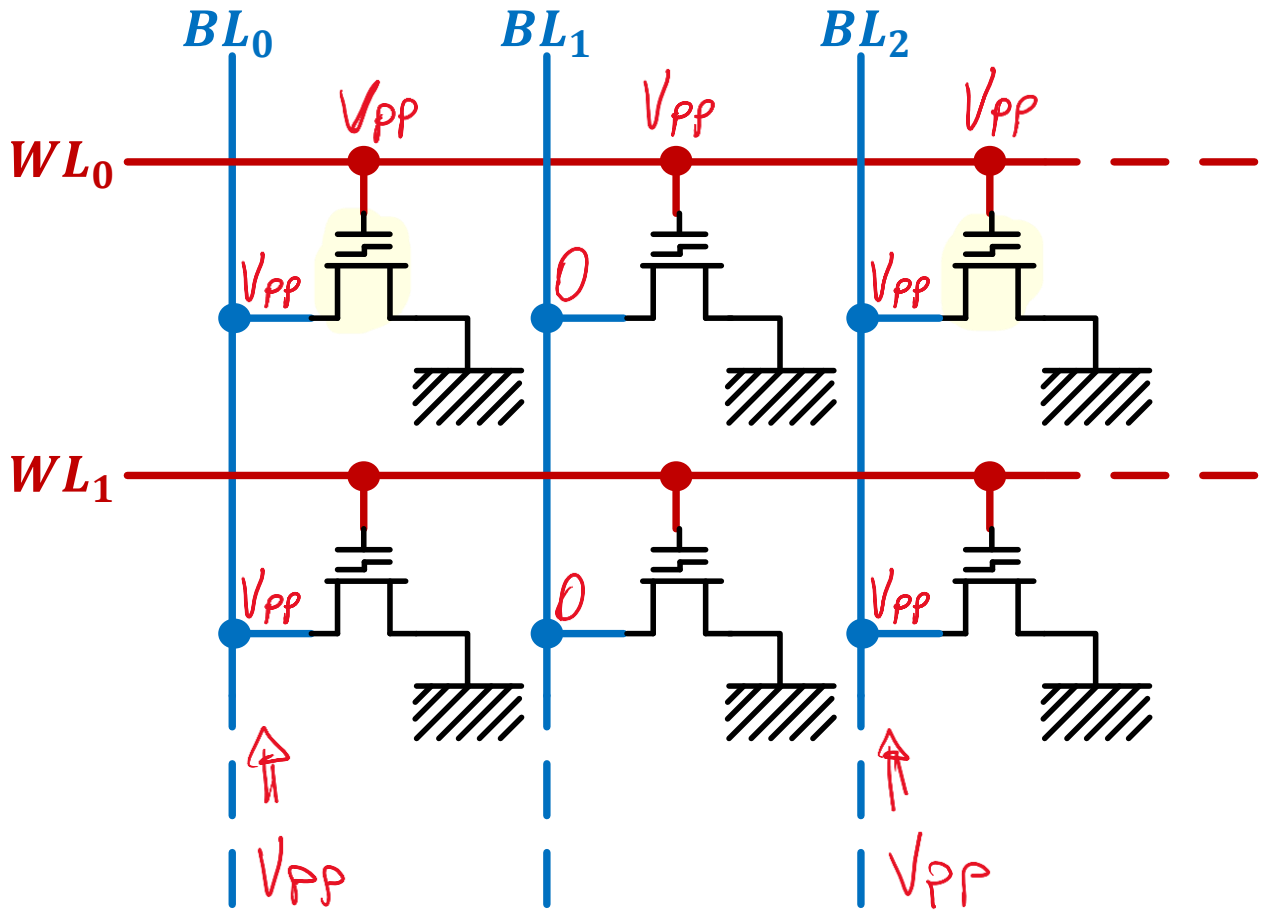
EEPROM (or E²PROM)

- Architecture (ideal)
  - Programming cell (0,1)
    - $V_G = WL_0 = V_{PP}$
    - $V_D = BL_1 = 0$



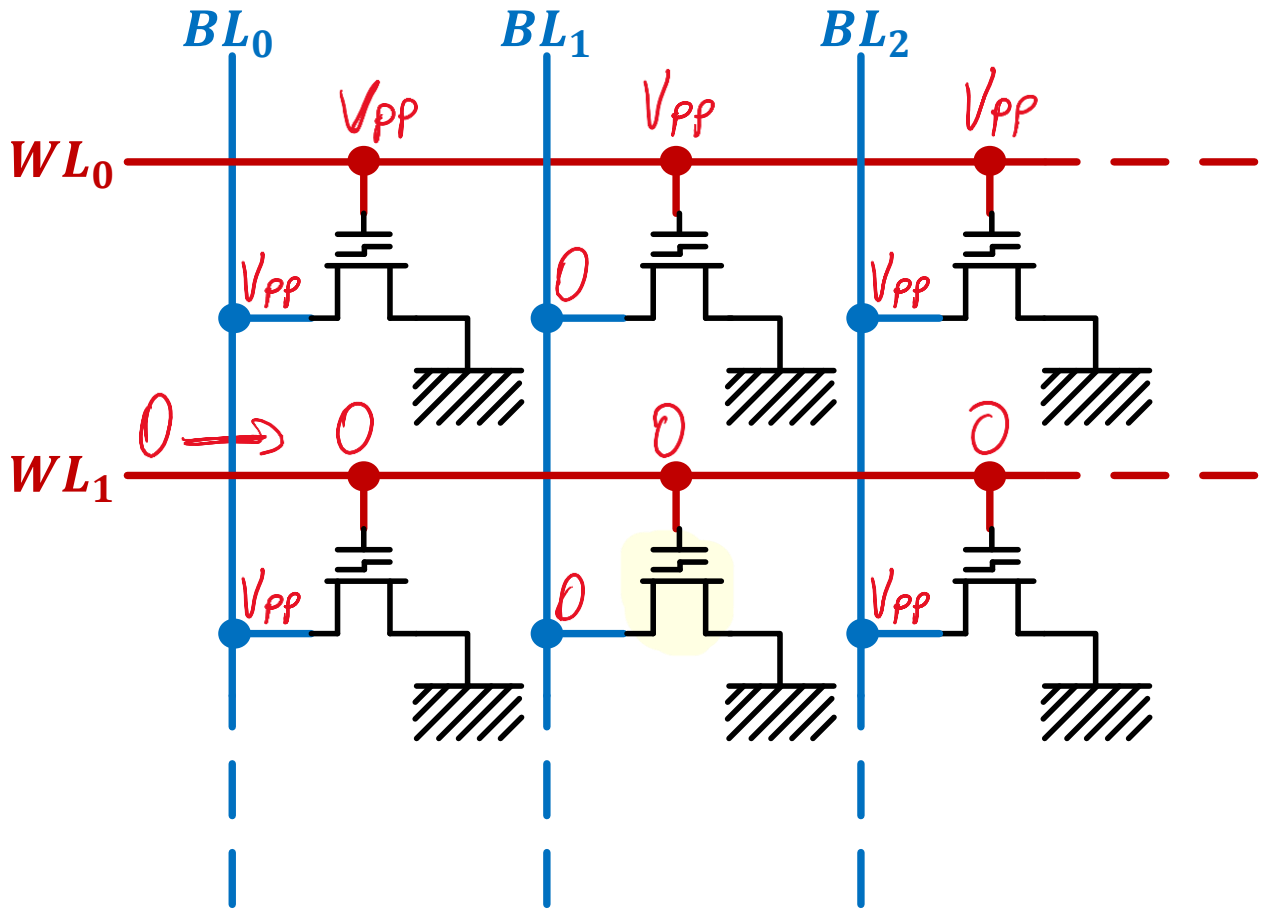
EEPROM (or E²PROM)

- Architecture (ideal)
  - Programming cell (0,1)
    - $V_G = WL_0 = V_{PP}$
    - $V_D = BL_1 = 0$
  - All other $BL_j = V_{PP}$
    - Otherwise the other cells in the same WL_0 are programmed !



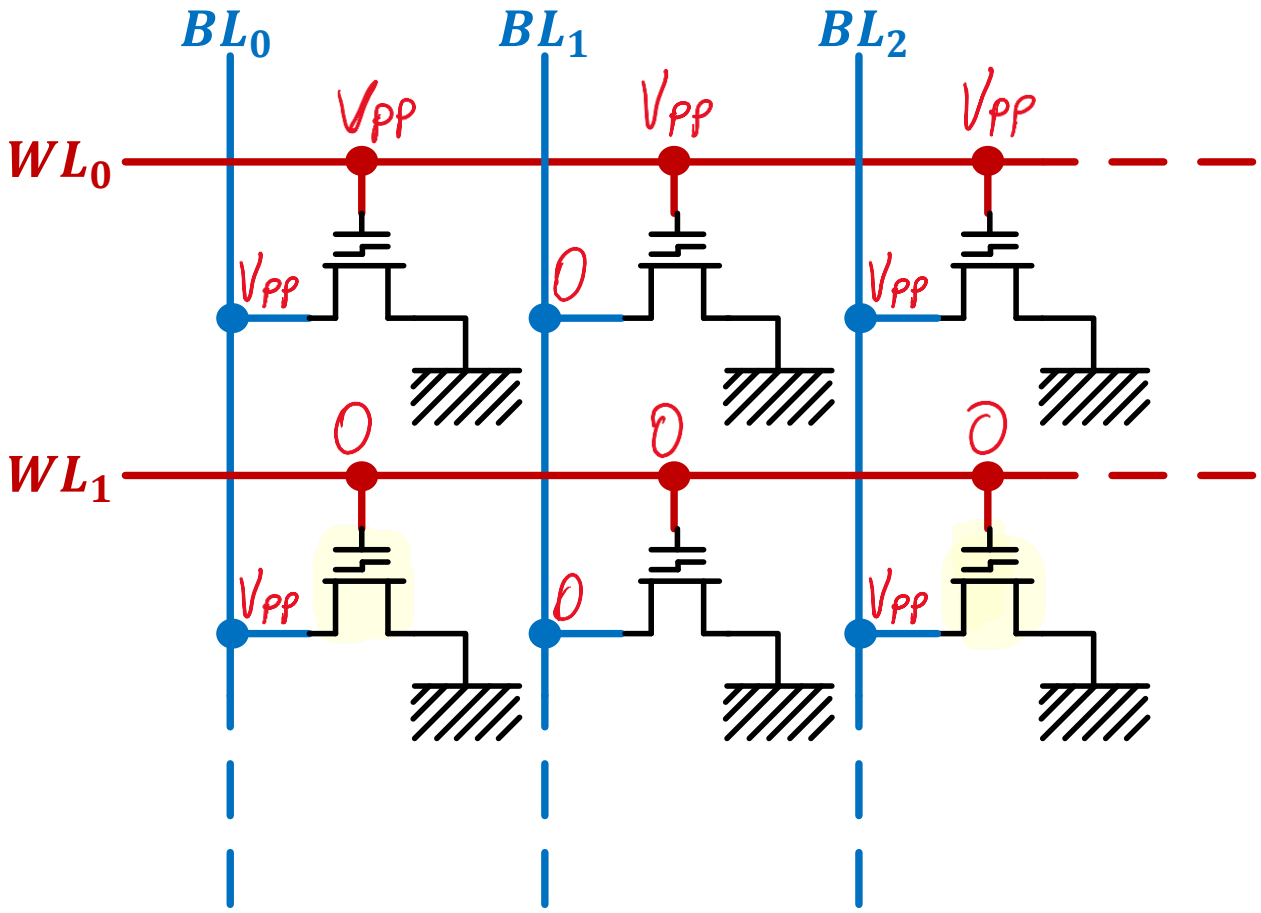
EEPROM (or E²PROM)

- Architecture (ideal)
  - Programming cell (0,1)
    - $V_G = WL_0 = V_{PP}$
    - $V_D = BL_1 = 0$
  - All other $BL_j = V_{PP}$
    - Otherwise the other cells in the same WL_0 are programmed !
  - All other $WL_i = 0$
    - Otherwise the other cells in the same BL_1 are programmed !



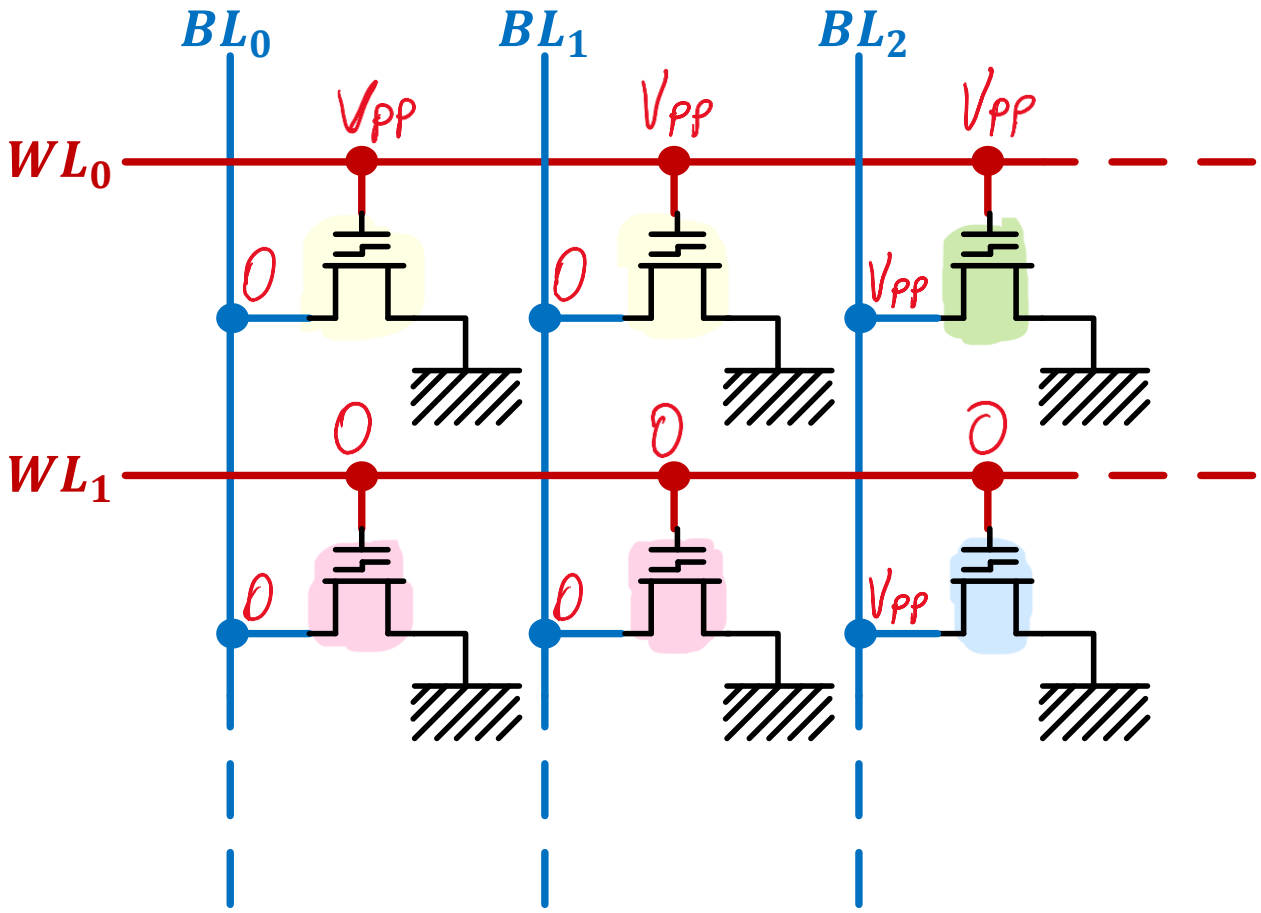
EEPROM (or E²PROM)

- Architecture (ideal)
  - Programming cell (0,1)
    - $V_G = WL_0 = V_{PP}$
    - $V_D = BL_1 = 0$
  - All other $BL_j = V_{PP}$
    - Otherwise the other cells in the same WL_0 are programmed !
  - All other $WL_i = 0$
    - Otherwise the other cells in the same BL_1 are programmed !
  - But … **ERASE DISTURB !!!**



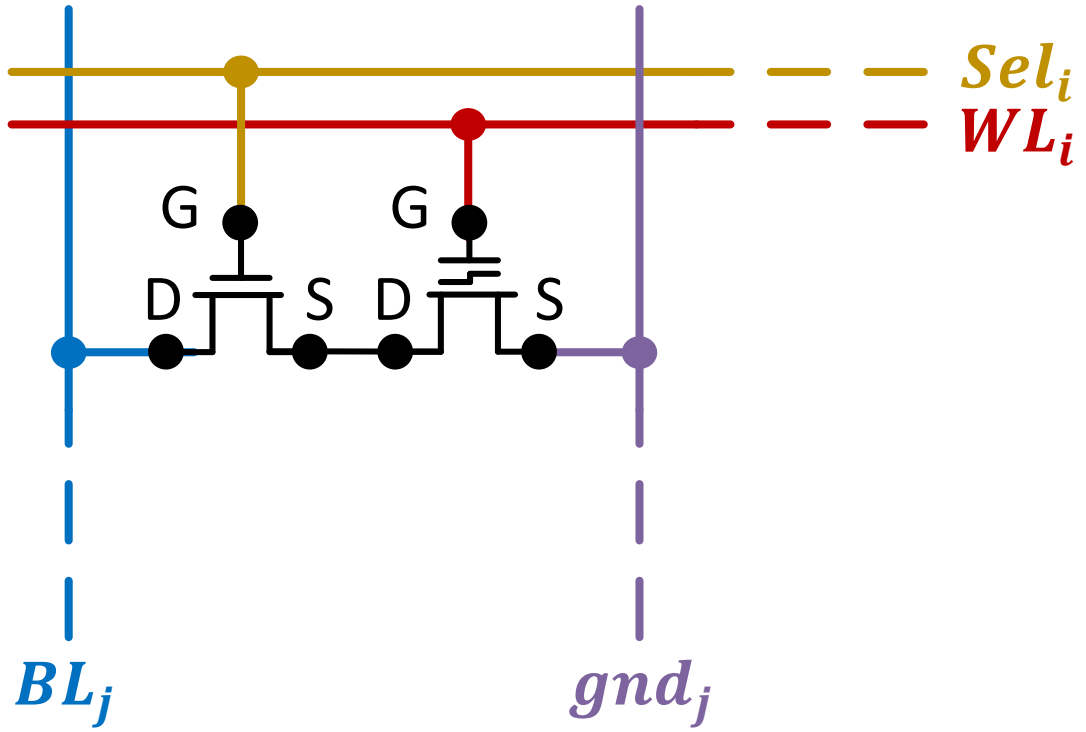
EEPROM (or E²PROM)

- Architecture (ideal)
  - Similarly, **WRITE DISTURB !!!**
  - Erasing cell (1,2)
    - $V_G = WL_1 = 0$
    - $V_D = BL_2 = V_{PP}$
  - All other $BL_j = 0$
    - Otherwise the other cells in the same WL_1 are erased !
  - All other $WL_i = V_{PP}$
    - Otherwise the other cells in the same BL_2 are erased !



EEPROM (or E²PROM)

- Architecture (real) – To solve the problems
  - Memory cell
    - 2 transistors
      - 1x FLOTOX
      - 1x n-MOS (pass transistor)



EEPROM (or E²PROM)

Just for the sake of completeness
- **It won't be part of the exam!!**

- Architecture (real) – To solve the problems
  - How should be driven

Operation	WL_i	Sel_i	BL_j	gnd_j	$WL_{i'}$	$Sel_{i'}$	$BL_{j'}$	$gnd_{j'}$
Program (WL_i,BL_j)	V_{CC}	V_{CC}	0	Z	0	0	V_{PP}	Z
Erase (WL_i,BL_j)	0	V_{CC}	V_{PP}	Z	V_{PP}	0	0	Z
Read (WL_i,BL_j)	V_{CC}	V_{CC}	Q	0	0	0	--	0

  - Notes
    - $\boldsymbol{WL_{i'}}$, $\boldsymbol{Sel_{i'}}$, $\boldsymbol{BL_{j'}}$, and $\boldsymbol{gnd_{j'}}$ indicate, respectively, all other WL, Sel, BL, and gnd different from WL_i, Sel_i, BL_j , and gnd_j
    - Z = high impedance

Thank you for your attention

Luca Crocetti
(luca.crocetti@unipi.it)