1980s

nMOS - only n-type devices

- **enhancement mode**
  - $V_T > 0$

- **depletion mode**
  - $V_T < 0$

- Inverter:
  - high resistance
  - $\text{OUT} = \overline{\text{IN}}$

- **effective resistance**
  - $R_{pu}$
  - $R_{pd} = \frac{r}{4} R_{pu}$

- Asymmetric, rise time $\approx 4 \times$ fall time

- Static power consumption for logic 0 output:

  - On (weakly)
  - OFF (current)

- Great for array design

- n-input NAND, n+1 transistors

(1)
pseudo-nMOS

true nMOS

CMOS imitating nMOS

narrow (min. width, poss. length > min)
high resistance
always on, pulling up.

V. good imitation in CMOS of nMOS

differences: nWell for pullup
pullup needs ground connection.

VDD

nWell

rest on n network

GND

Not big problem in PLAs.

3 components

VDD

S

S

END

P

GND

GND

NC

PD
n-input NOR

A_1 + A_2 + A_3 + ... + A_n

D = \quad D = \quad D =

VDD

Programmable Logic Array

NOR (AND)

NOR (OR)

FSM

CMOS
Running Example

\[ P_1 + P_2 = X = \overline{A \cdot B} + C \]
\[ P_3 + P_4 = Y = \overline{A} + B \cdot \overline{C} \]
\[ P_5 + P_1 = Z = \overline{A \cdot B \cdot C} + A \cdot B \]

\[ P_1 = A \cdot \overline{B} \]
\[ P_2 = C \]
\[ P_3 = \overline{A} \]
\[ P_4 = B \cdot \overline{C} \]
\[ P_5 = A \cdot \overline{B} \cdot \overline{C} \]

Simple approach is shown - many optimisation techniques works with nMOS, pseudo-nMOS does not work so well with "proper" CMOS Feedback with clocks gives finite state machines pseudo-nMOS is not used much because of static power consumption - but it leads to the next idea.
New idea

2 phase per clock cycle:

**COMPUTE**  - stable inputs
   - performing computation

**PRECHARGE**  - unstable input
   - pre-charging output to 1

! Let's build - nice idea!

[Diagram of a circuit with two phases showing alternating n-type and p-type dynamic gates.]

- Precharge to 1 when \( \phi = 0 \)
- Precharge to 0 when \( \bar{\phi} = 0 \)