



Hardware - PLD, CPLD

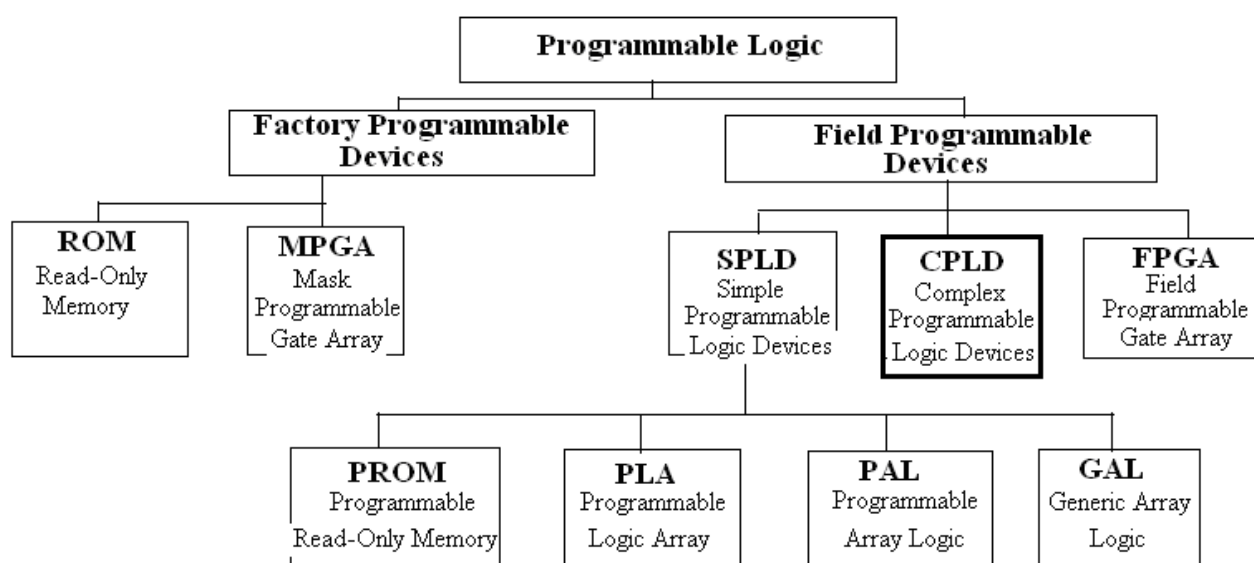
[Mise à jour le 10/8/2022]

- **Ressources**
 - [Circuit logique programmable](#)

1. Définition

Un **circuit logique programmable de type Field Programmable Devices**, ou **réseau logique programmable**, est un **circuit intégré logique** qui peut être reprogrammé (**caractérisé**) après sa fabrication.

2. Classification



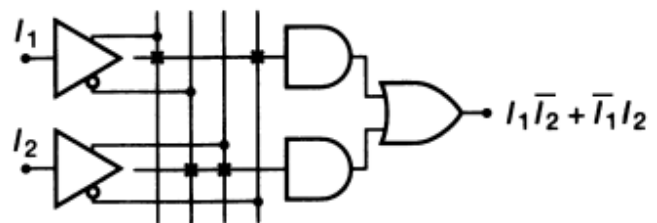
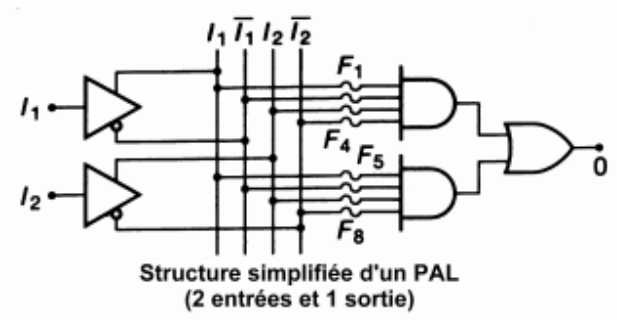
Ce type de composant électronique est communément désigné par différents sigles anglais dont :

- **FPGA** (field-programmable gate array, réseau de portes programmables in situ) ;
- **PLD** (programmable logic device, circuit logique programmable) ;
- **EPPLD** (erasable programmable logic device, circuit logique programmable et effaçable) ;
- **CPLD (complex programmable logic device, circuit logique programmable complexe)** ;
- **PAL** (programmable array logic, réseau logique programmable) ;
- **PLA** (programmable logic array, réseau logique programmable).

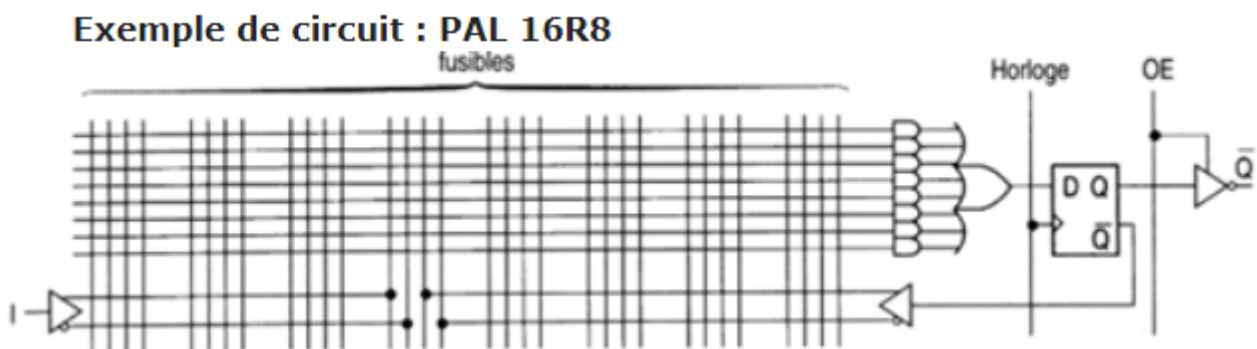
3. Organisation

Les **réseaux logiques programmables** sont des circuits composés de **nombreuses cellules logiques élémentaires** et **bascules logiques** librement interconnectables.

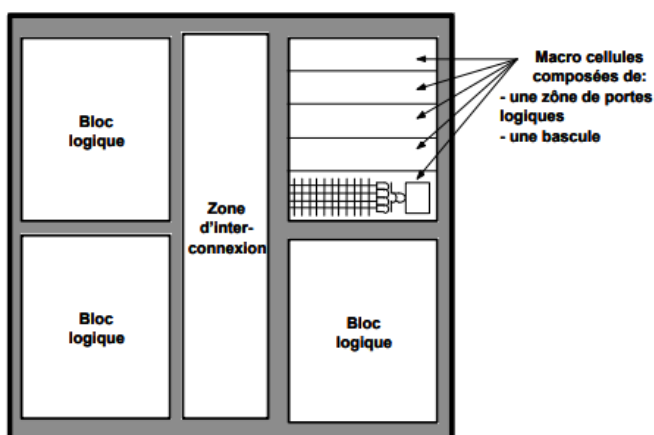
- **PAL (réalisation d'un OU-Exclusif)**



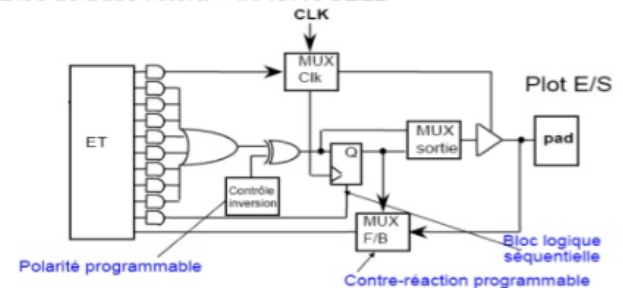
Exemple de circuit **PAL16R8**



- **CPLD**



Bloc de base Altera = MACROCELL



4. Programmation



Contrairement à un microprocesseur, **un circuit logique programmable n'exécute aucune ligne de code**. Un code de programmation est cependant nécessaire pour **assembler les opérateurs logiques** qui le composent.

La programmation des PLD modernes (FPGA / CPLD) passe généralement par un compilateur basé sur un langage de programmation de type langage de description matériel (ou « HDL » pour Hardware Description Language) comme le **ABEL**¹⁾. Pour faciliter la programmation, il existe aussi des langages de plus haut niveau. Les deux plus connus sont le **VHDL**²⁾ ("V" pour "Very high speed") et le **VERILOG**³⁾.

Exemple : comparateur logique

Un comparateur logique est un circuit logique auquel sont soumis en entrée deux mots sous forme binaire, et qui détermine lequel des deux nombres correspondants est le plus grand, le plus petit ou s'ils sont égaux.

Code VHDL d'un comparateur 1 bit

- **Description structurelle**

*.vhd

```
--Importation des bibliothèques
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- Description de l'interface externe
entity COMPAREUR1bit is port(
    A,B : IN STD_LOGIC;
    S : OUT STD_LOGIC;
end COMPAREUR1bit;

-- Description structurelle
architecture EQU_LOG of COMPAREUR1bit
begin
    S <= not(A xor B);
```

```
end EQU_LOG;
```

• Description fonctionnelle

*.vhd

```
-- Seule la partie description du code ci-dessus est modifiée
architecture COMPORT of COMPAREUR1bit
begin
    S <= '1' when A=B else '0'; -- C'est intuitif, concis et clair !
end COMPORT;
```

Code VHDL d'un comparateur 2 bits

• Description structurelle

*.vhd

```
--Importation des bibliothèques
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

-- Description de l'interface externe
entity COMPAREUR2bits is port(
    A,B : IN STD_LOGIC_VECTOR (1 downto 0); -- A et B sont des mots de
2bits : A = A1A0 et B = B1B0
    S : OUT STD_LOGIC;
end COMPAREUR2bits;

-- Description structurelle
architecture EQU_LOG of COMPAREUR2bits
begin
    S <= ( not(A(1)) and not(A(0)) and not(B(1)) and not(B(0)))
        or (not(A(1)) and A(0) and not(B(1)) and B(0))
        or (A(1) and not(A(0)) and B(1) and not(B(0)))
        or (A(1) and A(0) and B(1) and B(0));
end EQU_LOG;
```

• Description fonctionnelle

*.vhd

```
-- Seule la partie description du code ci-dessus est modifiée
architecture COMPORT of COMPAREUR2bits
begin
    S <= '1' when A=B else '0'; -- Identique au comparateur 1bit et
```

```
généralisable à n bits !!!  
end COMPORT;
```

Les descriptions comportementales sont toujours recommandées, sauf raison majeure de densité ou de performance après synthèse.

5. Utilisation

Pour de **moyennes séries**, les composants programmables sont en général préférés aux **ASIC**. Les PLD peuvent être **programmés par l'utilisateur** (bureau d'étude, chaîne d'assemblage) alors que les ASIC qui sont configurés lors de leur fabrication (ils ne peuvent être que simulés par le bureau d'étude) sont réservés à des productions de très grande série ne serait-ce qu'en raison de leur coût de développement élevé.

1)

ABEL, acronyme de Advanced Boolean Expression Language, est un langage de programmation informatique utilisé dans le domaine de l'électronique, pour notamment la programmation des PLD. Il a été créé en 1983 par Data I/O Corporation, à Redmond, Washington.

2)

VHDL est un langage de description de matériel destiné à représenter le comportement ainsi que l'architecture d'un système électronique numérique. Son nom complet est **VHSIC Hardware Description Language**.

3)

Le **Verilog**, de son nom complet **Verilog HDL** est un langage de description matériel de circuits logiques en électronique, utilisé pour la conception d'ASICs et de FPGAs. Le sigle anglais HDL - Hardware Description Language- signifie Langage de Description du Matériel.

From:

<https://webge.fr/dokuwiki/> - **WEBGE Wikis**

Permanent link:

<https://webge.fr/dokuwiki/doku.php?id=materiels:pld&rev=1660204350>

Last update: **2022/08/11 09:52**

