CRESITT INDUSTRIE

Centre de Ressources
Technologiques en Électronique

Intelligence Artificielle & Composants électroniques
Retour d'expérience d'implémentation sur FPGA

Christophe ALAYRAC – 12/10/2023 – 1.0

Réf du document : DT_PPT_CA_IA&Composants2031012.odp

Le CRT CRESITT est soutenu par :

L'action de diffusion technologique est cofinancée par l'Union européenne.
L'Europe s'engage en région Centre-Val de Loire avec le Fonds européen de développement régional.

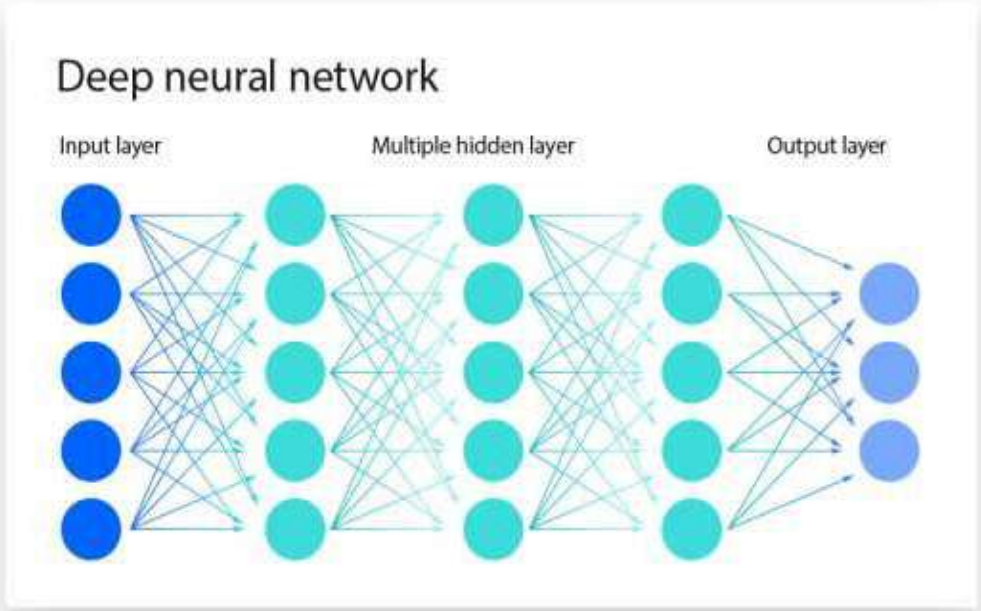
SOMMAIRE

- Aperçu rapide des réseaux d’IA
- Focus sur le CNN
- Application chiffres manuscrits
- Les ressources nécessaires
- Les ressources du FPGA
- L’implémentation
- Les performances

Aperçu rapide

Source : https://www.ibm.com/fr-fr/topics/neural-networks

Les réseaux de neurones tentent d'imiter le cerveau humain, en combinant l'informatique et les statistiques pour résoudre des problèmes courants dans le domaine de l'intelligence artificielle



couches nodales
Interconnexion des nœuds
possède un poids et un seuil

Si out > seuil sortie utilisée

données d'entraînement

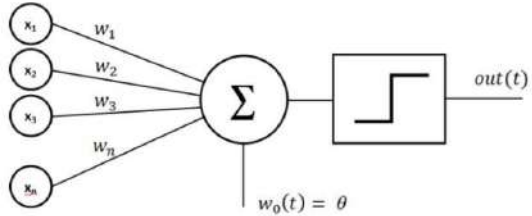
$\sum w_i x_i + biais = w_1x_1 + w_2x_2 + w_3x_3 + biais$

$sortie = f(x) = 1 \text{ si } \sum w_1x_1 + b >= 0 \text{ ; } 0 \text{ si } \sum w_1x_1 + b < 0$

Quelques types

Source : https://www.ibm.com/fr-fr/topics/neural-networks

Le **perceptron** est le plus ancien réseau de neurones, créé par Frank Rosenblatt en 1958.

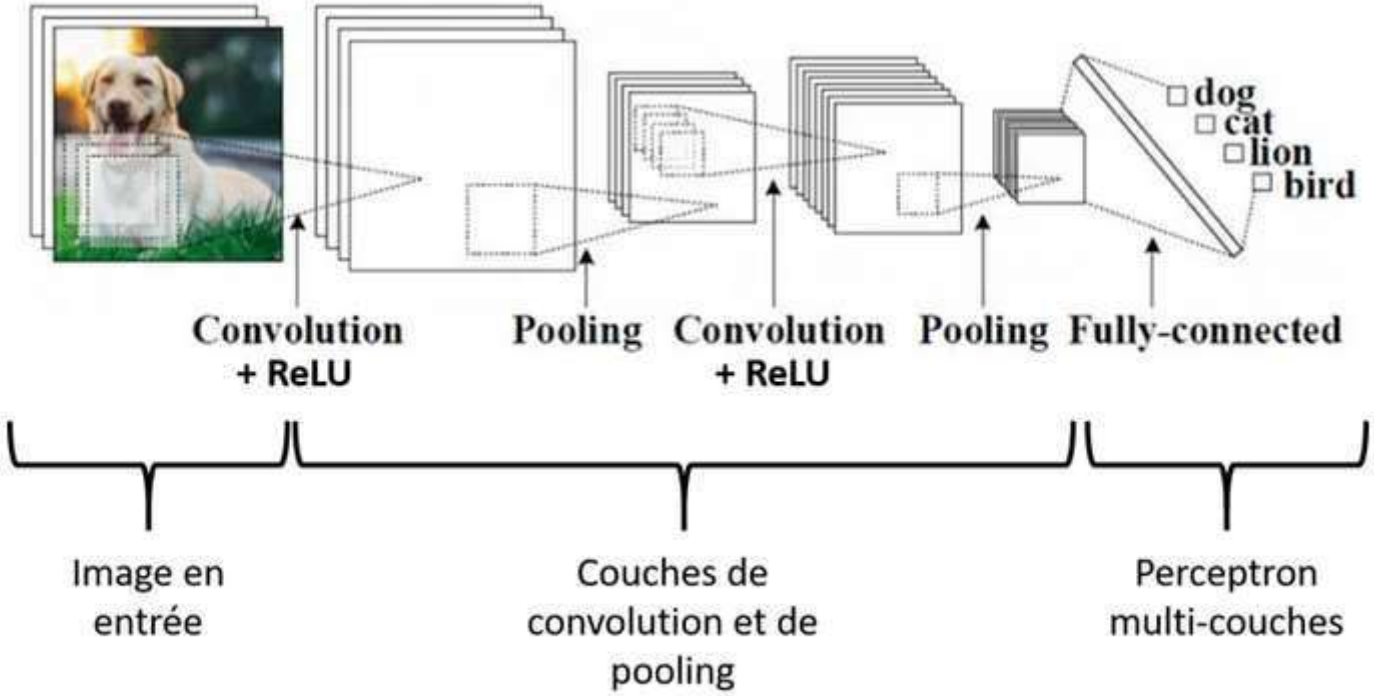


- A propagation avant (perceptrons multicouches (MLP)) :
  → une couche d'entrée, une ou plusieurs cachées et une de sortie.
  → constitués de neurones sigmoïdes (fonction en forme de S)
  car la plupart des problèmes du monde réel sont non-linéaires.

- convolutifs (CNN)
  → similaires aux réseaux à propagation avant,
  → reconnaissance d'images, de formes et/ou la vision par ordinateur.
  → algèbre linéaire, multiplication des matrices, identification motifs dans une image.

- récurrents (RNN)
  → boucles de rétroaction.
  → données de séries chronologiques
  → prédictions résultats futurs

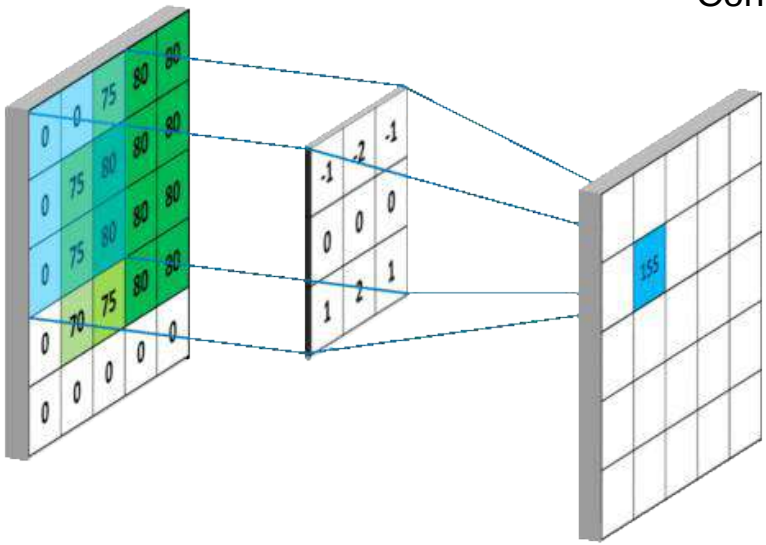
Focus sur le CNN

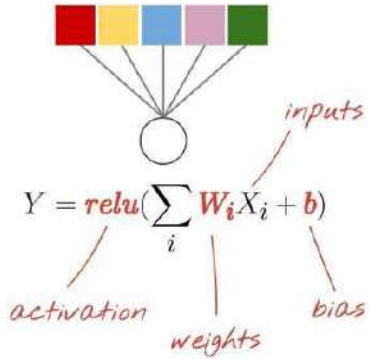


source : https://www.aspexit.com/

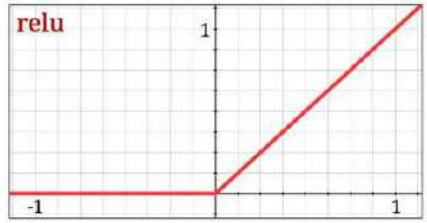
Focus sur le CNN

Convolution



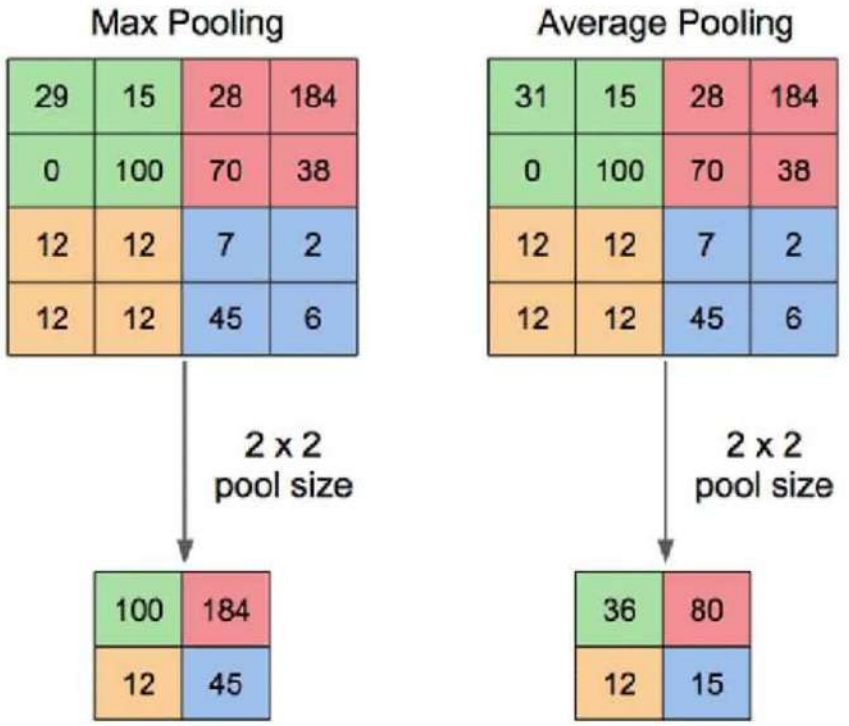


Rectified Linear Unit

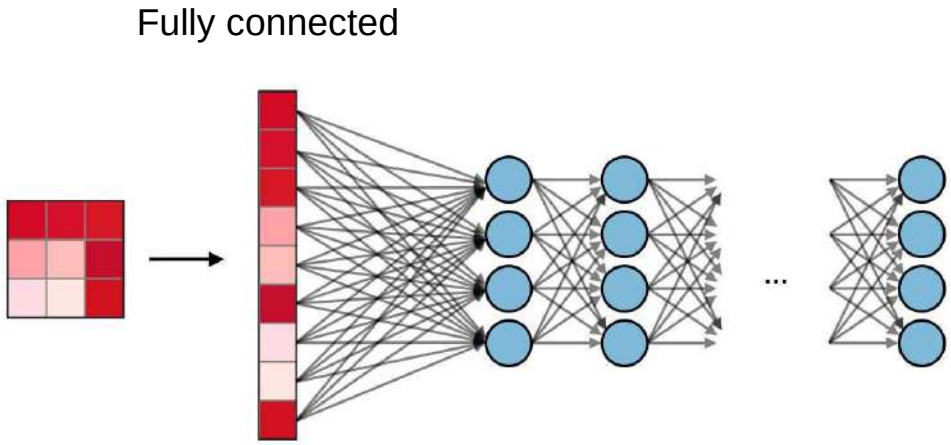


Source https://yannicksergeobam.medium.com

Focus sur le CNN

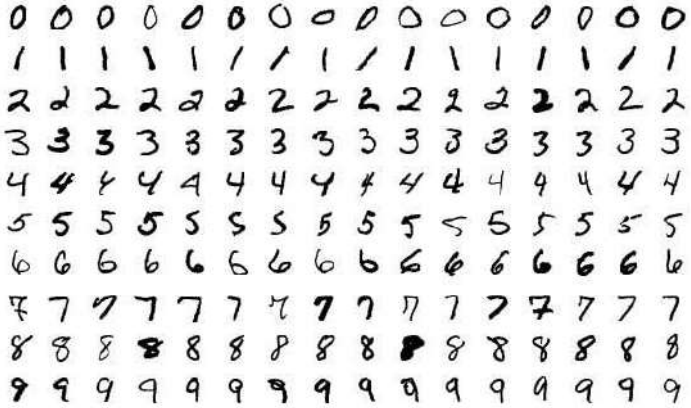


Source: https://www.researchgate.net/



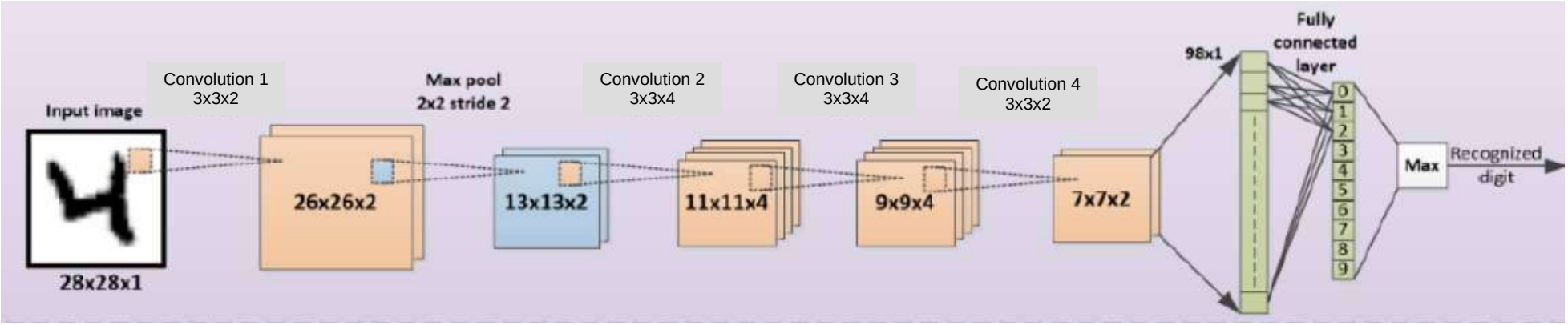
Source: https://stanford.edu/

Application chiffre manuscrit

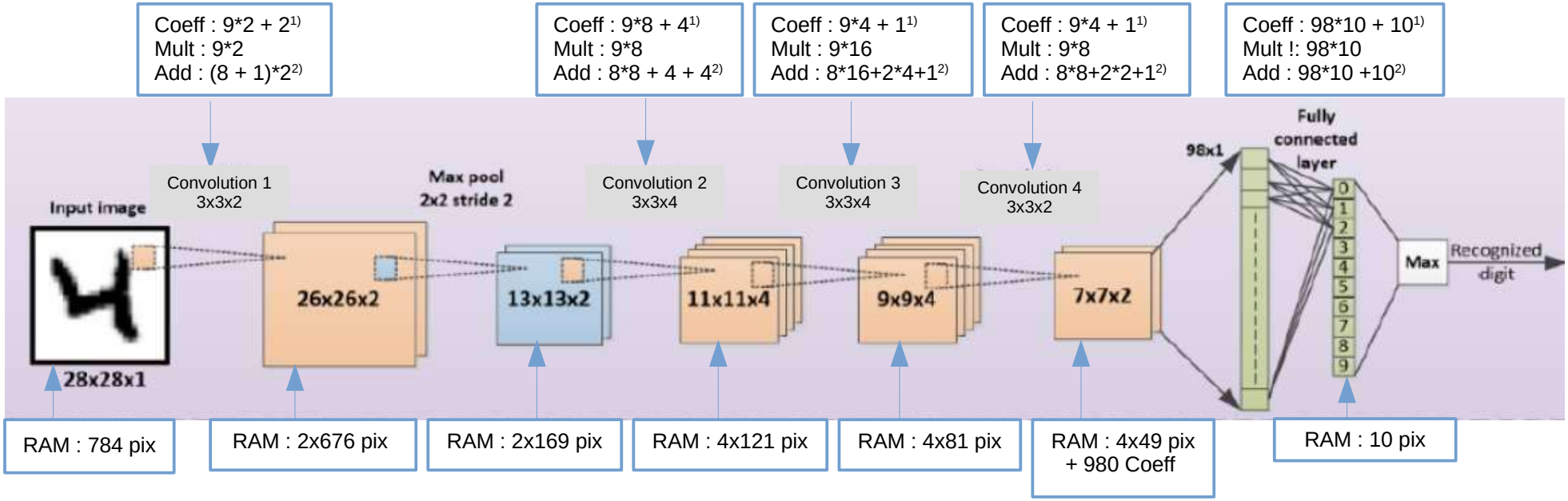


Base de données MNIST
Image 28x28 pixels
60000 images d'apprentissage
10000 images de test

Sorties 10 classes (0 à 9)
Pas de classe chiffre absent



Les ressources nécessaires

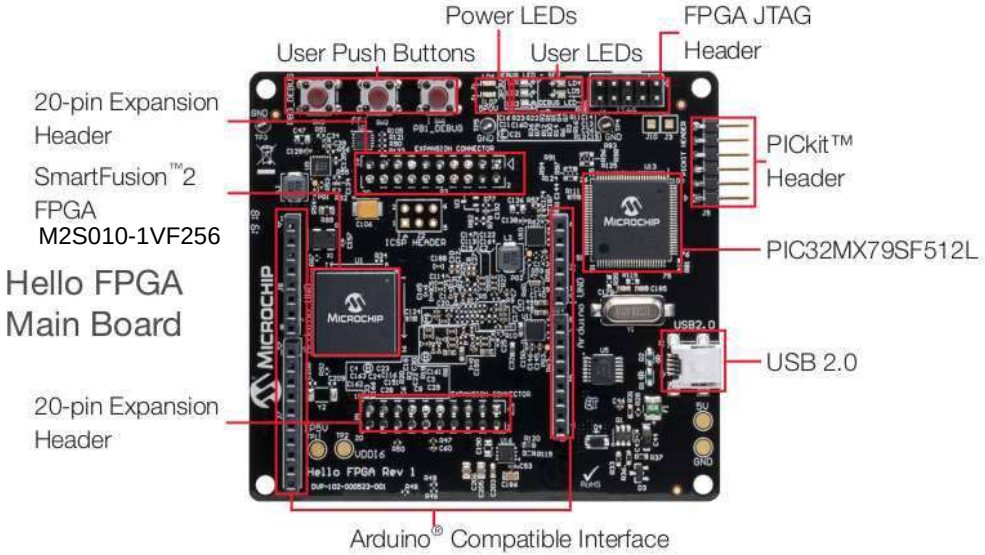


Note :

[1)]Coeff : Conv + biais

*[2)]Accumulateur pix_bit*2+9*

Kit Hello FPGA Mircrosemi



Camera Sensor Board
640x480
RGB565

Hello FPGA Main Board

LCD Board

M2S-HELLO-FPGA-KIT

Les ressources FPGA

SmartFusion2 Devices — SOC : Système On Chip

SmartFusion2 Devices	Features	M2S005	M2S010	M2S025	M2S050	M2S060	M2S090	M2S150
Logic/DSP	Maximum logic elements (4LUT + DFF)	6,060	12,084	27,696	56,340	56,520	86,184	146,124
	Mathblocks (18 × 18)	11	22	34	72	72	84	240
	Fabric interface controllers (FICs)	1			2	1		2
	PLLs and CCCs	2			6			8
	Security	AES256, SHA256, RNG				AES256, SHA256, RNG, ECC, PUF		
MSS	Cortex-M3 + instruction cache	Yes						
	eNVM (KB)	128	256				512	
	eSRAM (KB)	64						
	eSRAM (KB) non-SECDED	80						
	CAN, 10/100/1000 Ethernet, HS USB	1 each						
	Multi-mode UART, SPI, I2C, timer	2 each						
Fabric memory	LSRAM 18K blocks	10	21	31	69		109	236
	uSRAM 1K blocks	11	22	34	72		112	240
	Total RAM (kbits)	191	400	592	1,314		2,074	4,488
High-speed	DDR controllers (count × width)	1 × 18			2 × 36	1 × 18		2 × 36
	SERDES lanes	0	4		8	4		16
	PCIe endpoints	0	1		2			4
User I/O	MSIO (3.3 V)	115	123	157	139	271	309	292
	MSIOD (2.5 V)	28	40		62	40		106
	DDRIO (2.5 V)	66	70		176	76		176
	Total user I/Os	209	233	267	377	387	425	574

Les ressources FPGA

Figure 10 • **Simplified Functional Block Diagram for LSRAM**

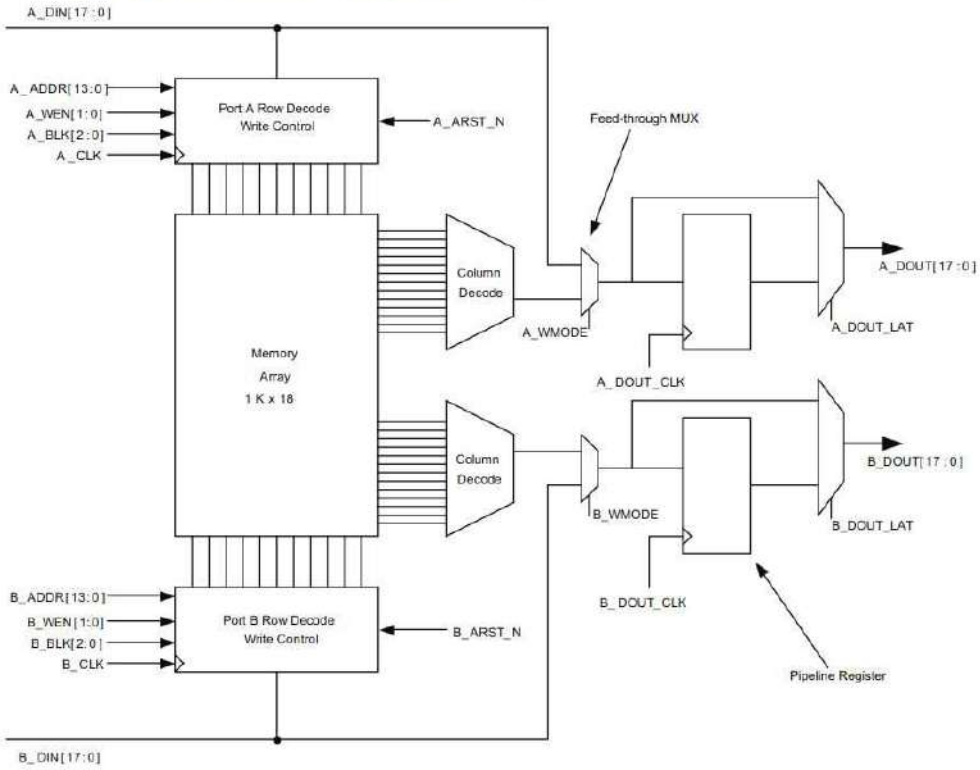


Table 7 • **Read/Write Operation Selection**[1, 2]

Depth x Width	A_WEN/B_WEN	Operation
16K x 1 8K x 2 4K x 4 2K x 8 2K x 9 1K x 16 1K x 18	00	Read operation
16K x 1 8K x 2 4K x 4 2K x 8 2K x 9 1K x 16 1K x 18	1	Write operation
512 x 32 (Two-port write-Port B)	A_WEN[1:0] = “11” B_WEN[1:0] = “11”	Write [31:0]
512 x 36 (Two-port write-Port B)	B_WEN[1:0] = “11” A_WEN[1:0] = “11”	Write [35:0]

1. In dual-port mode, every port reads when the corresponding write enable (A_WEN/B_WEN) is "00" and corresponding port select (A_BLK/B_BLK) is active.

Les ressources FPGA

Figure 24 • **Simplified Functional Block Diagram of µSRAM**

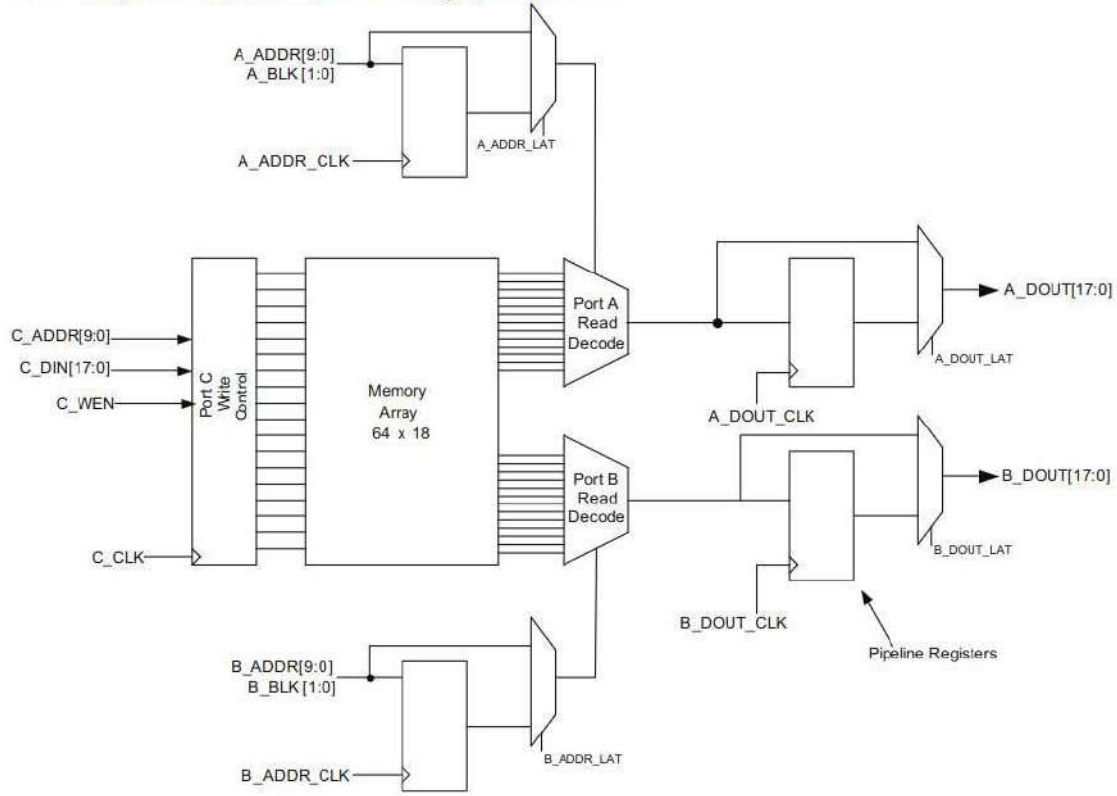


Table 28 • **Data Input Buses Used and Unused Bits**

	C_DIN Ecriture	
Depth x Width	**Used Bits**	**Unused Bits (to be grounded)**
1K x 1	[0]	[17:1]
512 x 2	[1:0]	[17:2]
256 x 4	[3:0]	[17:4]
128 x 8	[7:0]	[17:8]
128 x 9	[8:0]	[17:9]
64 x 16	[16:9] [7:0]	[17] [8]
64 x 18	[17:0]	None

Table 29 • **Data Output Buses Used and Unused Bits**

	A_DOUT/B_DOUT Lecture	
Depth x Width	**Used Bits**	**Unused Bits**
1K x 1	[0]	[17:1]
512 x 2	[1:0]	[17:2]
256 x 4	[3:0]	[17:4]
128 x 8	[7:0]	[17:8]
128 x 9	[8:0]	[17:9]
64 x 16	[16:9] [7:0]	[17] [8]
64 x 18	[17:0]	

MACC

Figure 36 • **Functional Block Diagram of the Math Block**

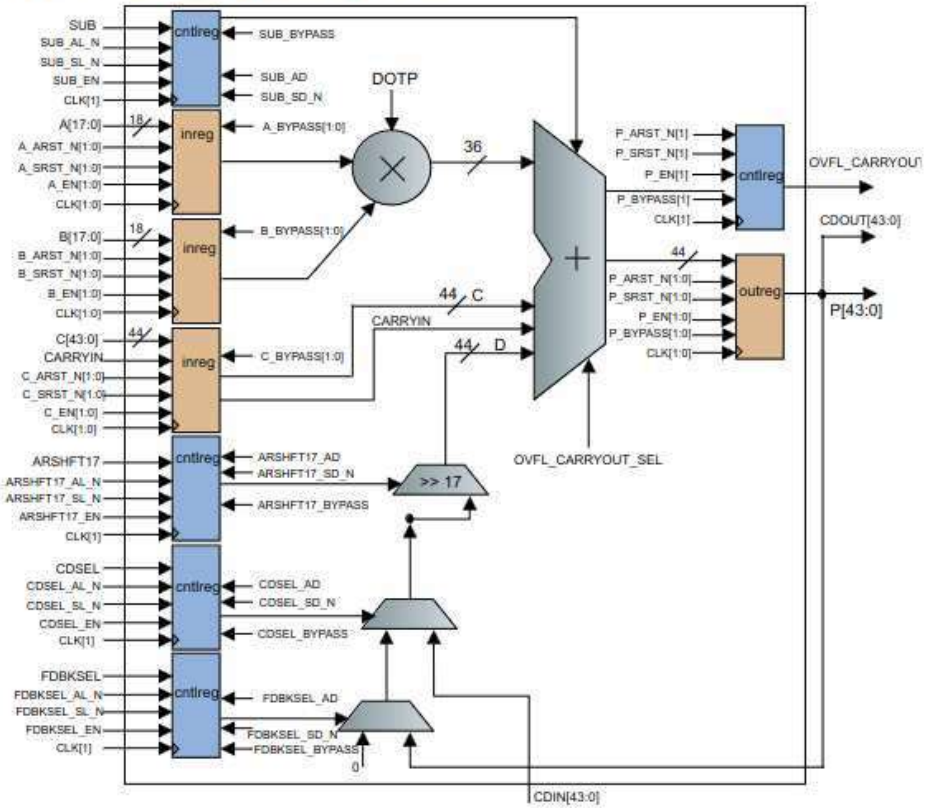


Figure 37 • **Functional Block Diagram of the Math Block in Normal Mode**

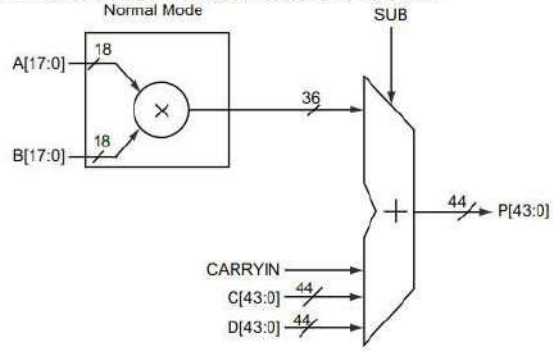
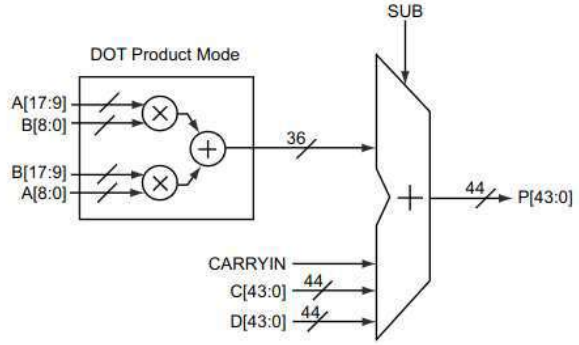
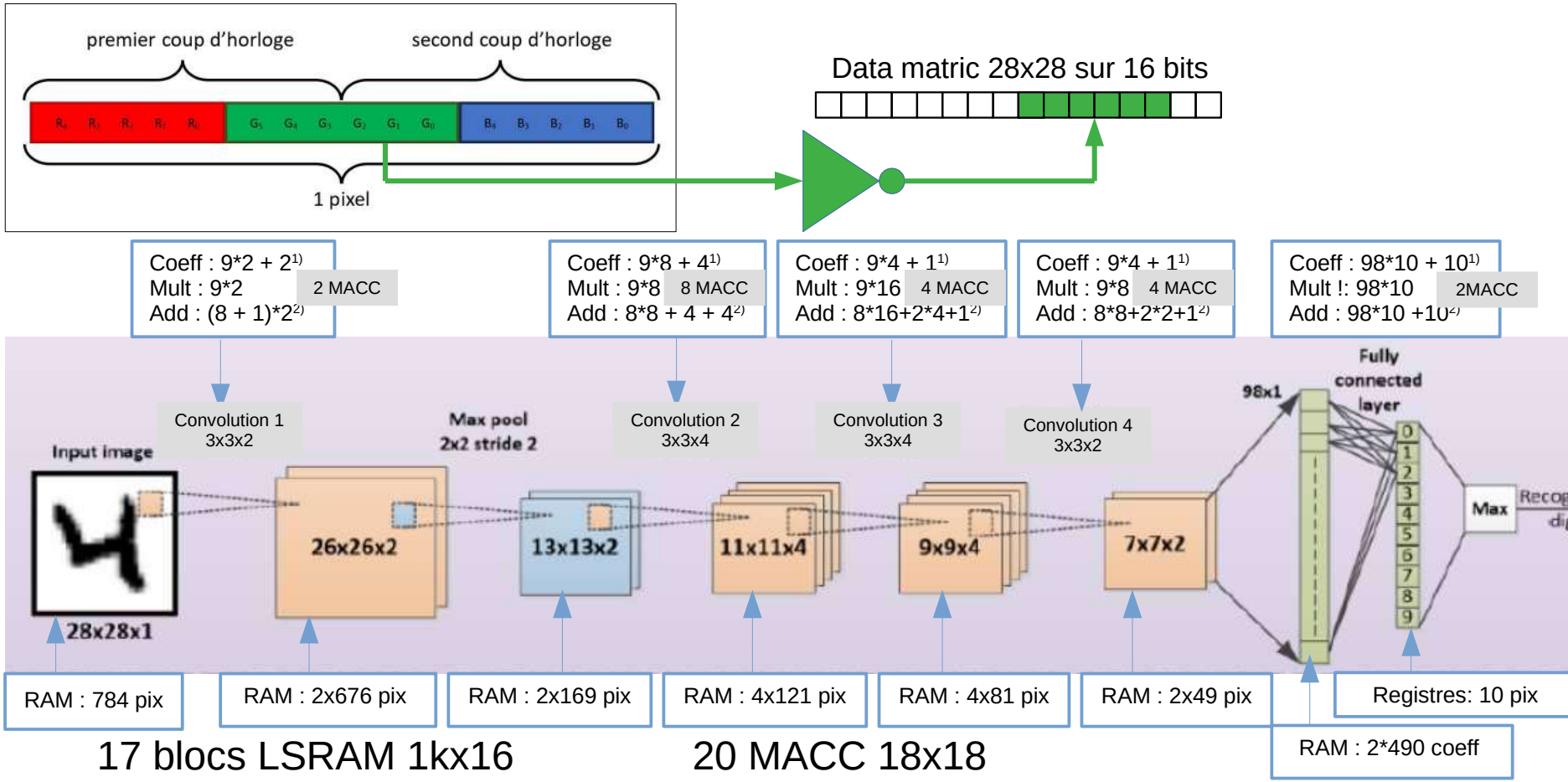


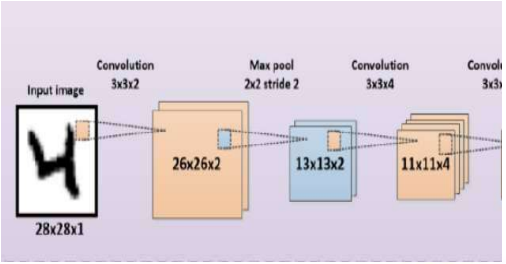
Figure 38 • **Functional Block Diagram of the Math Block in DOTP Mode**

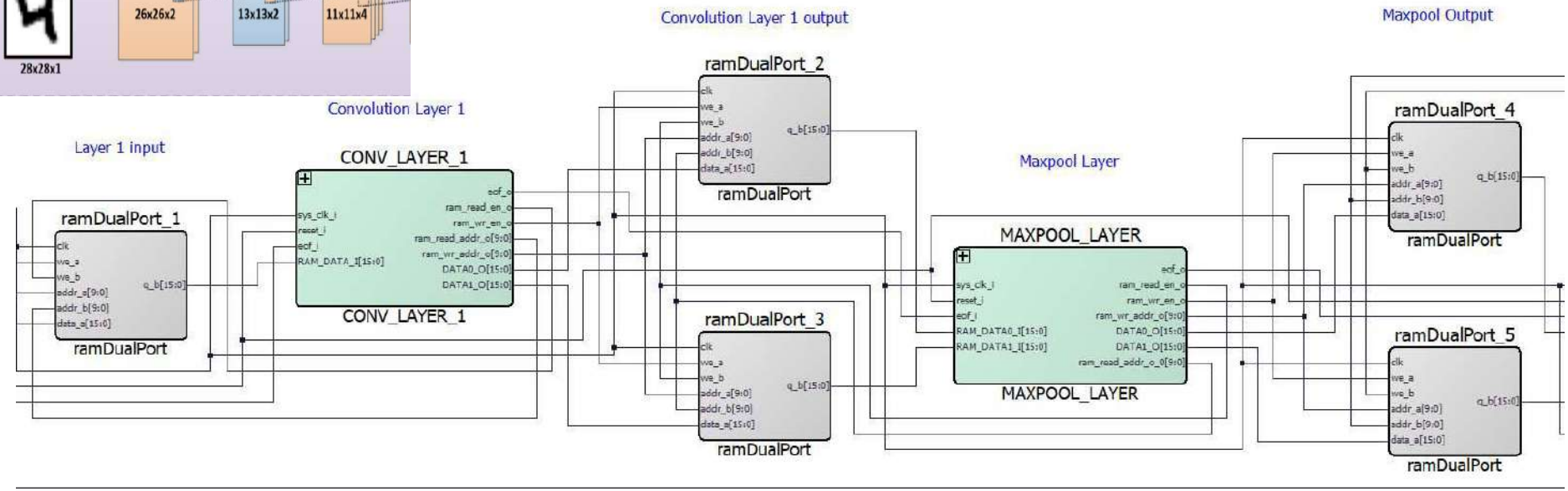


Les ressources FPGA

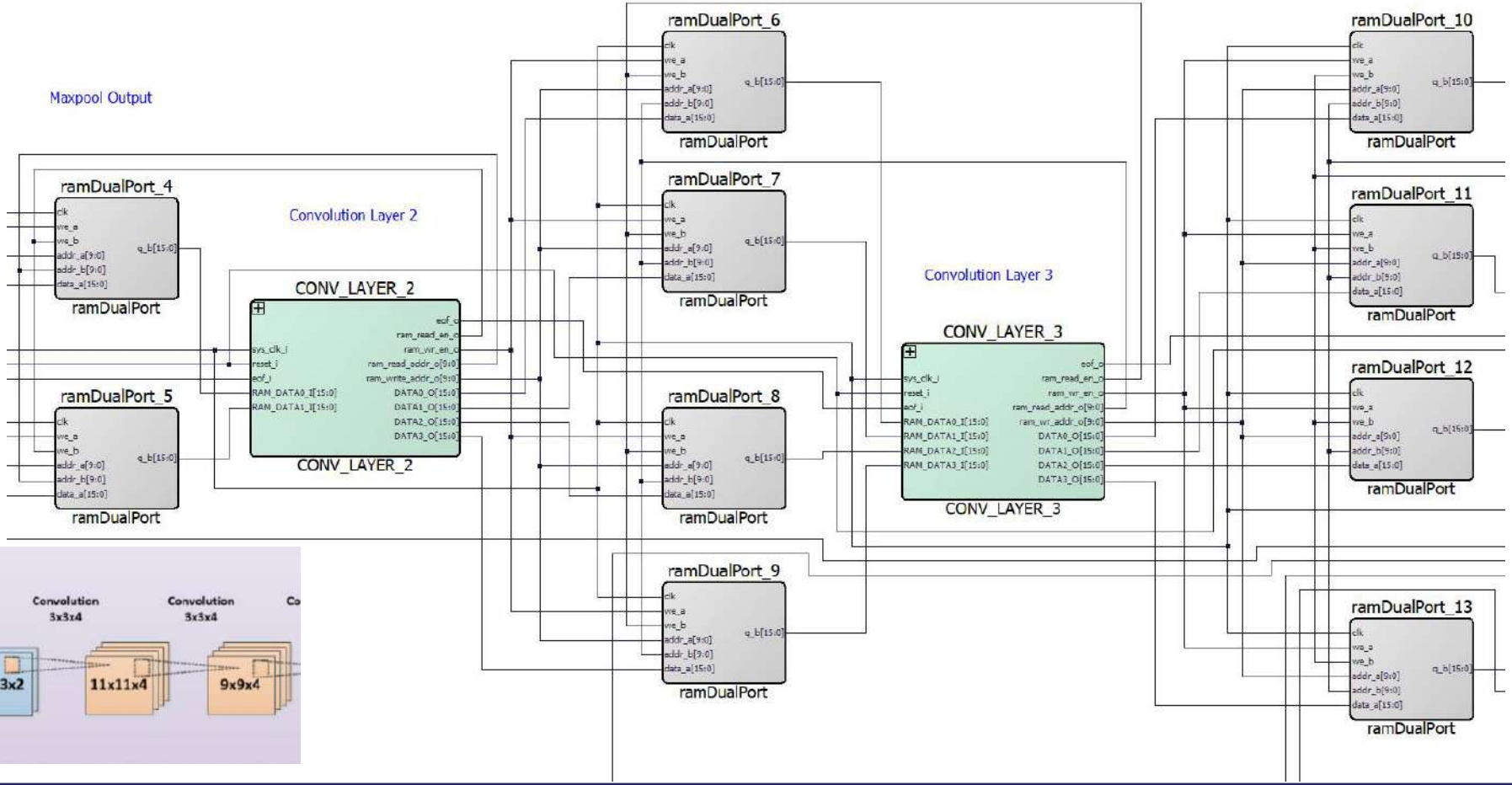


L'implémentation

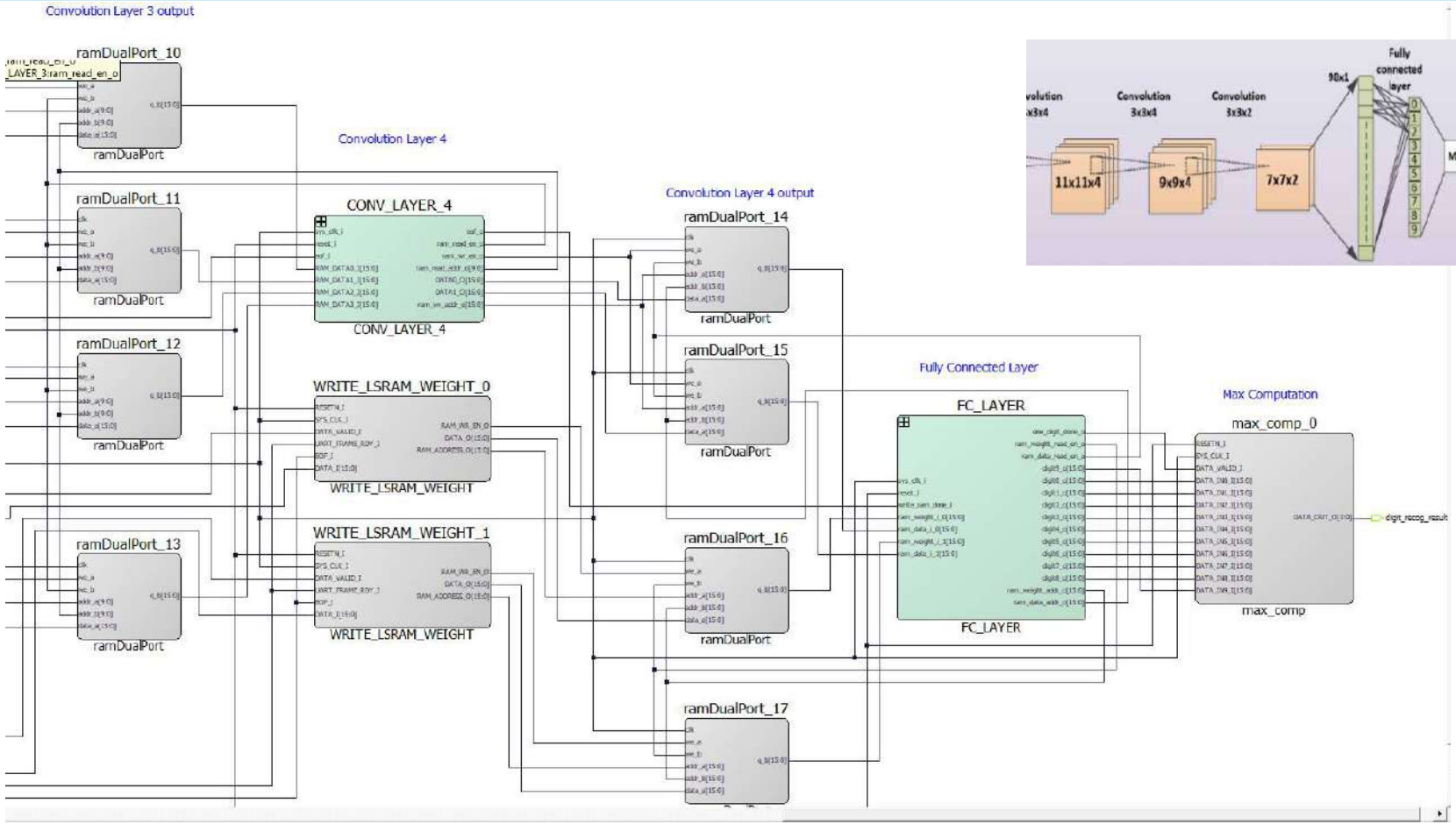




L'implémentation

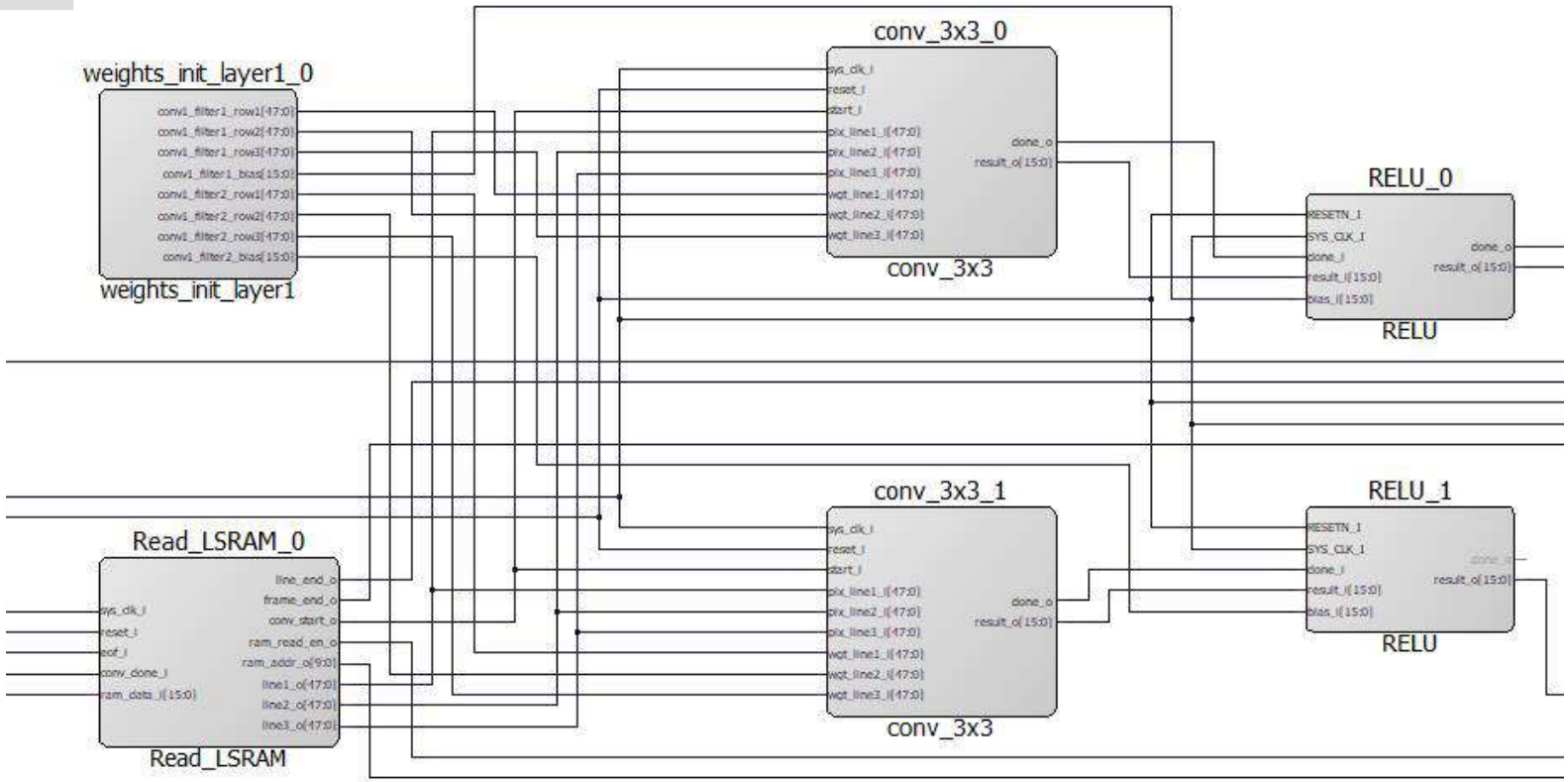


L'implémentation



L'implémentation

Convolution 1
3x3x2



L'implémentation

Coefficients Convolution 1

```
==============================
-- Top level output port assignments
--============================================================
==============================
    conv1_filter1_row1  <= x"E8B40AA227C9";
    conv1_filter1_row2  <= x"E8C51ED91D50";
    conv1_filter1_row3  <= x"FAFA1F111C0D";
    conv1_filter1_bias  <= x"0002";

    conv1_filter2_row1  <= x"23460FDF05DD";
    conv1_filter2_row2  <= x"F345202C1CA9";
    conv1_filter2_row3  <= x"D9E0D613F0EF";
    conv1_filter2_bias  <= x"FFFD";
```

Implanté en dur dans le FPGA.

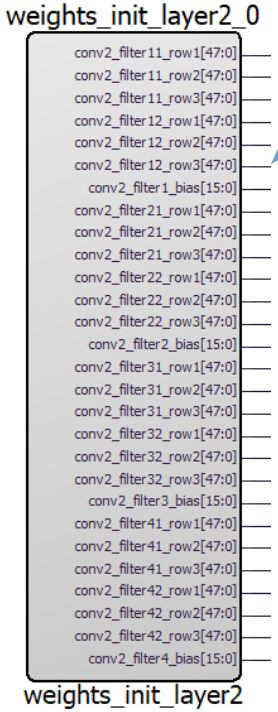
ReLu

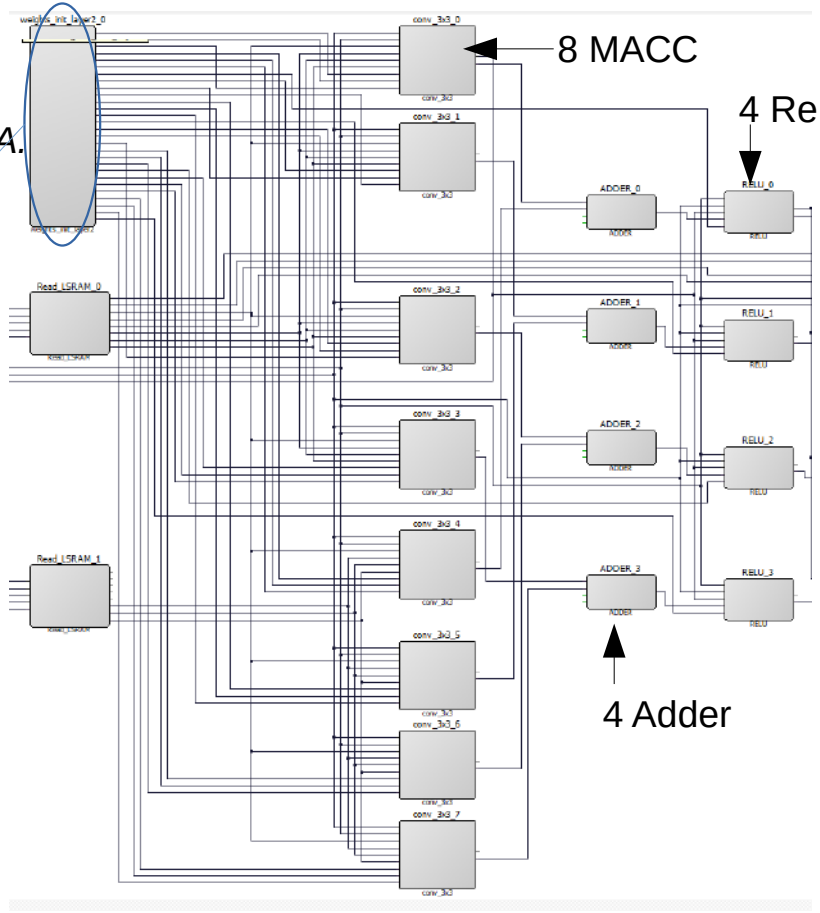
```
--============================================================
-- Top level output port assignments
--============================================================

result_o <= (OTHERS=>'0' )WHEN (s_result(15) = '1') ELSE s_result;

--============================================================
-- GENERATE blocks
--============================================================
--NA--

--============================================================
-- Asynchronous blocks
--============================================================
--NA--
--============================================================
-- Synchronous blocks
--============================================================
------------------------------------------------------------------
-- Name       : RAM_ADDRESS_GEN
-- Description: Generates the RAM address
------------------------------------------------------------------
RAM_ADDRESS_GEN:PROCESS (RESETN_I,SYS_CLK_I)
    BEGIN
        IF(RESETN_I = '0')THEN
            done_o          <= '0';
            s_result        <= (OTHERS=>'0');
        ELSIF(rising_edge(SYS_CLK_I))THEN
            done_o          <= done_i;
            s_result        <= result_i + bias_i;
        END IF;
    END PROCESS;
```

L'implémentation

Convolution 2
3x3x2

Coefficients Convolution 2
Implanté en dur dans le FPGA.

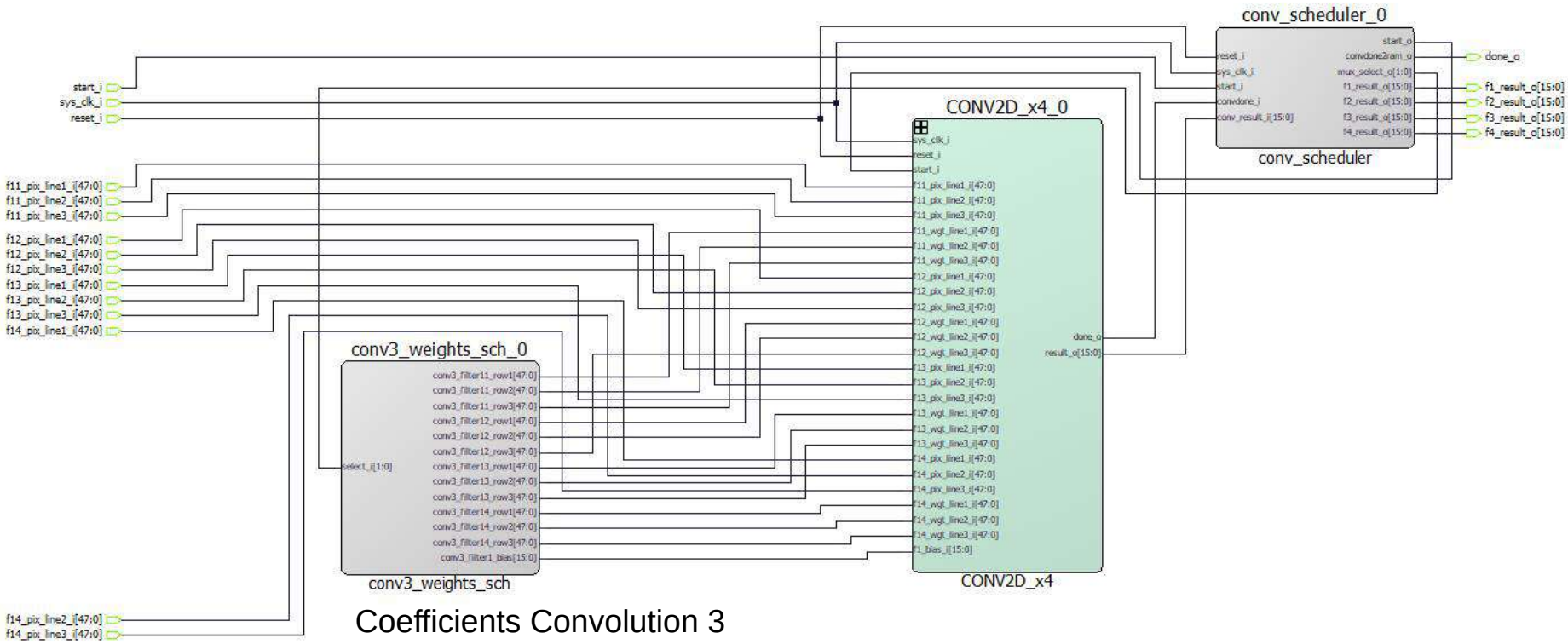




L'implémentation

Convolution 3
3x3x4

Séquenceur

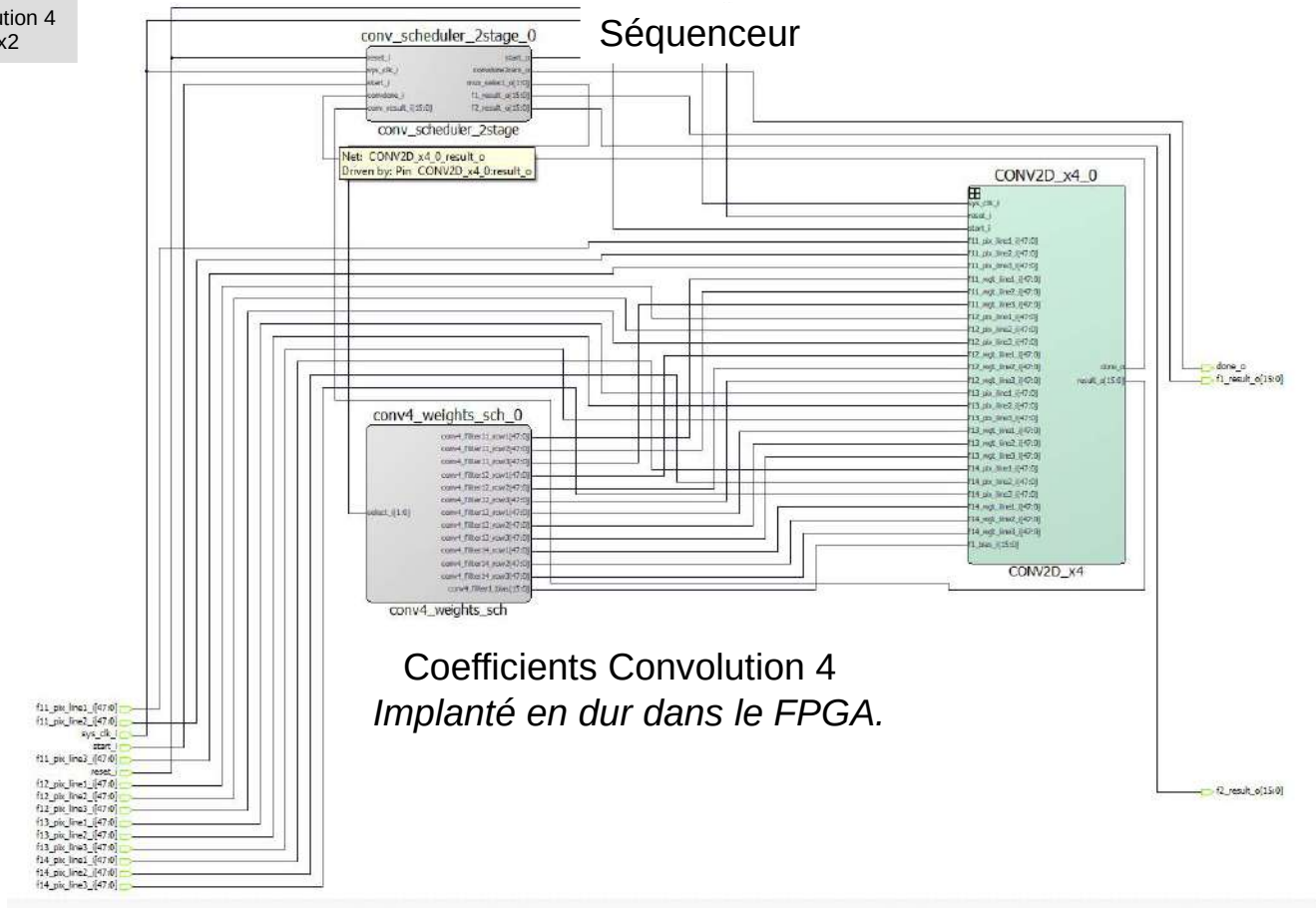


Coefficients Convolution 3

Implanté en dur dans le FPGA.

L'implémentation

Convolution 4
3x3x2



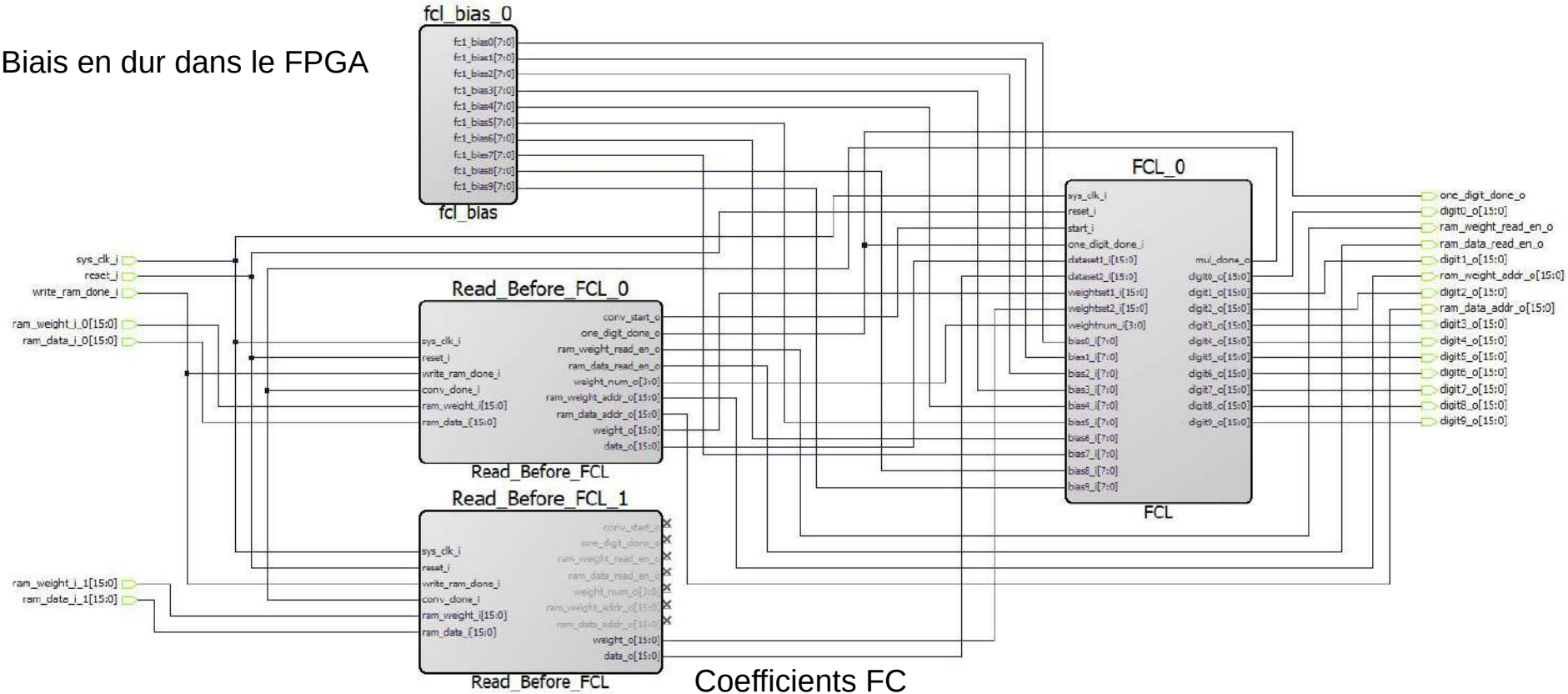
Séquenceur

Coefficients Convolution 4
Implanté en dur dans le FPGA.

L'implémentation

FC

Biais en dur dans le FPGA



Coefficients FC

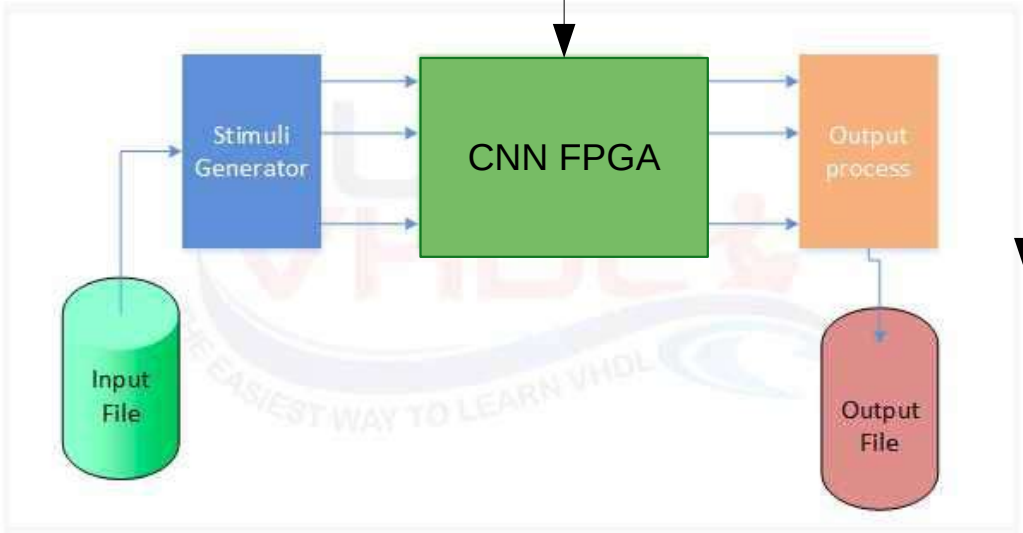
Implanté LSRAM, chargé par processeur ARM à l'init

L'implémentation

Table 2-1. Resource Utilization

Type	Used	Total	Percentage
4LUT	7203	12084	59.61
DFF	7168	12084	59.32
User I/O (single-ended)	37	138	26.81
RAM1K18	20	21	95.24
MACC	20	22	90.91

Les performances



python™

10000 images

Taux d'échec : 22,84 %

		Résultat du CNN									
		0	1	2	3	4	5	6	7	8	9
Résultat attendu	0	423	10	121	3	2	26	348	1	28	18
	1	856	68	7	14	68	1	70	14	36	1
	2	43	22	932	8	0	0	8	6	13	0
	3	1	0	4	983	0	4	0	4	9	5
	4	1	0	2	1	950	0	8	1	1	18
	5	3	0	0	10	0	846	13	1	11	8
	6	3	1	7	0	4	3	938	0	2	0
	7	257	7	28	7	1	0	0	696	4	28
	8	7	9	18	4	3	2	5	8	910	8
	9	7	0	1	6	6	3	0	12	4	970

Figure 12 : Tableau représentant la précision de l'algorithme de la carte FPGA

Les performances

	Raspberry PI 3		Ordinateur portable		FPGA
	pi camera	camera USB	webcam intégrée	camera USB	
FPS	2,77	2,75	29,88	28,66	30

Figure 13 : Tableau représentant les résultats des tests de performance

Autre exemple sur PolarFire
Microsemi_UG0943_CNN_Accelerator_IP_User_Guide.pdf

Conclusion

- ↘ Optimisation d'intégration en fonction des ressources disponibles
  → FPGA avec MACC et mémoire

- ↘ localisation des coefficients, des résultats intermédiaires
  →temps d'accès mémoire impact cadence du process

- ↘ troncature des coefficients & des résultats intermédiaires
  → dégradation performance sans précautions

Questions ?

?

Contacts

Christophe Alayrac

Directeur technique

christophe.alayrac@cresitt.com