

LABORATOIRE DE PHYSIQUE DES LASERS

Atelier d'électronique



DDS CARD AD9852 et AD9858 Description

Réf. : Cartes DDS AD9858 et AD9852 sur carte National Instrument
DIO 32 HS.

Equipe : GQD Strontium
Auteur : FABRICE WIOTTE

1. Description du projet et principe de fonctionnement du BUS
2. Description des principales fonctions de l'AD9852
3. Le schéma électronique et le PCB de la carte AD9852
4. Description des principales fonctions de l'AD9858
5. Le schéma électronique et le PCB de la carte AD9858
6. Présentation des composants logiques programmables (XC2C64A).
7. Les programmes associés aux composants logiques
8. La liste des composants des cartes AD9852 et AD9858
9. Quelques mesures sur le bus de données avec la carte AD9852

DESCRIPTION

1. Description du projet :

Pour une nouvelle expérience du laboratoire, nous avons développé de nouvelles cartes électroniques qui utilisent des composants DDS contrôlés par un ordinateur sur un bus de données numériques.

La carte numérique qui contrôle le bus est la carte DIO 32 HS de National Instrument. Ces sources RF programmables à base de technologie DDS seront mises en œuvre sur la nouvelle expérience des atomes froids de l'équipe Gaz Quantique Dipolaire pour conduire les AOM de l'expérience.

Pour contrôler les cartes, un programme simple sous LABVIEW a été développé.

La principale motivation de ce projet était de construire un système de contrôle pour l'expérience qui permet d'intégrer de nombreuses fonctions avec une interface utilisateur.

Les cartes DDS permettront de contrôler avec une bonne synchronisation les différents AOM.

Le bus de données :

Le bus est un bus parallèle de 25 lignes qui est transféré avec un câble plat de 50 lignes, de sorte que les lignes de masses et les lignes de signaux sont alternées. La vitesse du bus dépend des caractéristiques des cartes NI fournies soit de l'ordre de 1/10MHz soit 100ns.

Les 16 premières lignes du bus sont les lignes des données et les huit dernières lignes correspondent aux adresses d'appareils (255 en théorie mais 6 bits en pratique et donc 64 appareils adressables).

Enfin, la dernière ligne (ligne 25) est l'horloge (STROBE) qui contrôle le débit de données qui sont transmis sur le bus. Ce signal met à jour les données et les adresses sur le bus.

Ce signal d'horloge doit être présent sur le bus.

Programmation des DDS :

Les huit premières lignes de données transmises au DDS sont les DATA : D0-D7

Les huit secondes lignes transmises au DDS sont les ADRESSES : A0-A5 et A6-A7 pour les modes FSK et OSK de l'AD9852 et PS0 PS1 pour l'AD9858.

Les huit dernières lignes permettent le décodage d'adresses : decodeA2-decodeA7

Décode au maximum 64 cartes.

Les lignes decodeA0 et decodeA1, sont utilisées pour initialisées le DDS de tel sorte que :

A1 A0 = 00 : Master reset DDS

A1 A0 = 01 : Load data DDS

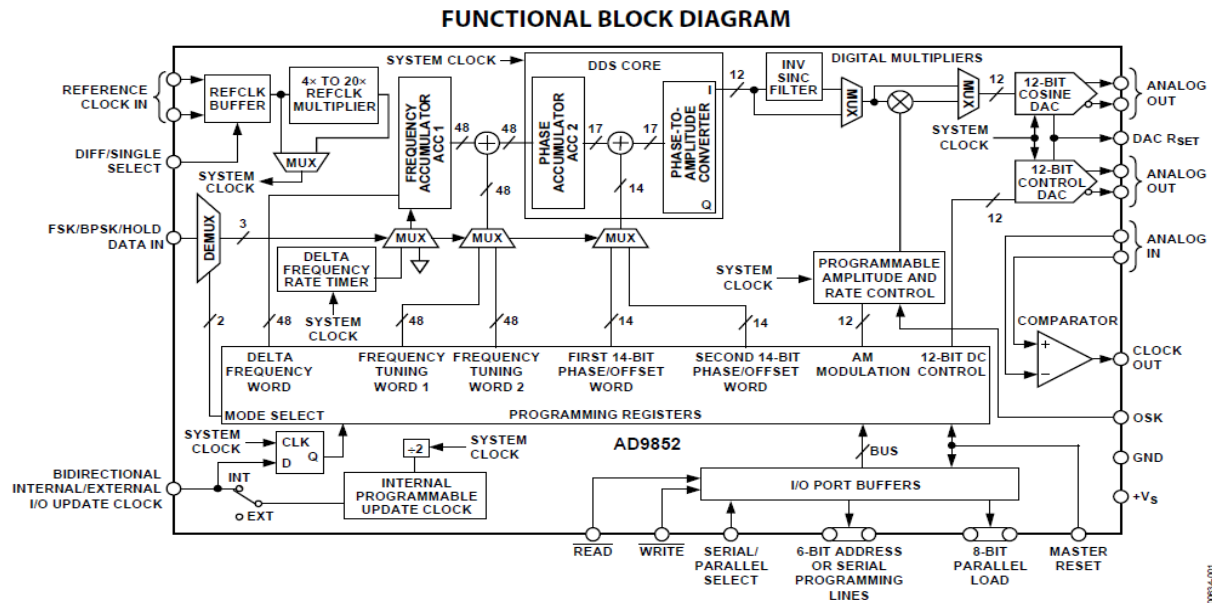
A1 A0 = 10 : I/O update DDS non utilisé si I/O Update interne.

A1 A0 = 11 : non utilisé

Description des principaux composants utilisés :

1. l'AD9852 CMOS 300MSPS COMPLETE DDS
2. l'AD9858 CMOS 1GSPS COMPLETE DDS
3. CPLDs XC2C64A XILINX

2. Diagramme fonctionnel de l'AD9852: L'AD9852



Principales fonctions:

REFERENCE CLOCK IN si l'horloge est externe = 20MHz et PLL activée x15 fréquence du system clock =300MHz. Sinon si PLL off CLOCK EXTERNE = 300MHz max

MASTER RESET bit actif sur front logic 1 initialise le DDS et met la sortie RF à zéro. pin 71 du DDS.

ANALOG OUT est la sortie RF principale 0dBm en général.

I/O UPDATE CLOCK est le bit de chargement des données dans le DDS (par défaut généré en interne et synchronisé sur system clock. Pin 20 du DDS)

FSK/BPSK/HOLD bit actif sur front logic 1 lance les rampes de fréquences en autres.

OSK bit actif sur front logic 1 lance les rampes d'amplitudes.

8-BIT PARALLEL LOAD bus 8bits pour chargement des données actif D0-D7 du DDS.

6-BIT ADDRESS bus 6bits pour chargement des adresses en mode parallèle. A0-A5 du DDS.

SERIAL/PARALLEL SELECT bit de sélection des modes de programmation du DDS: = 1 si chargement parallèle, 0 si chargement série. Pin 70 du DDS.

DIFF/SINGLE SELECT bit de sélection single ended or differential clock input. pin 64 du DDS.

WRITE OR WR/SCLK si mode Parallèle => bit de chargement I/O port buffers, actif sur front descendant. Pin 21 du DDS

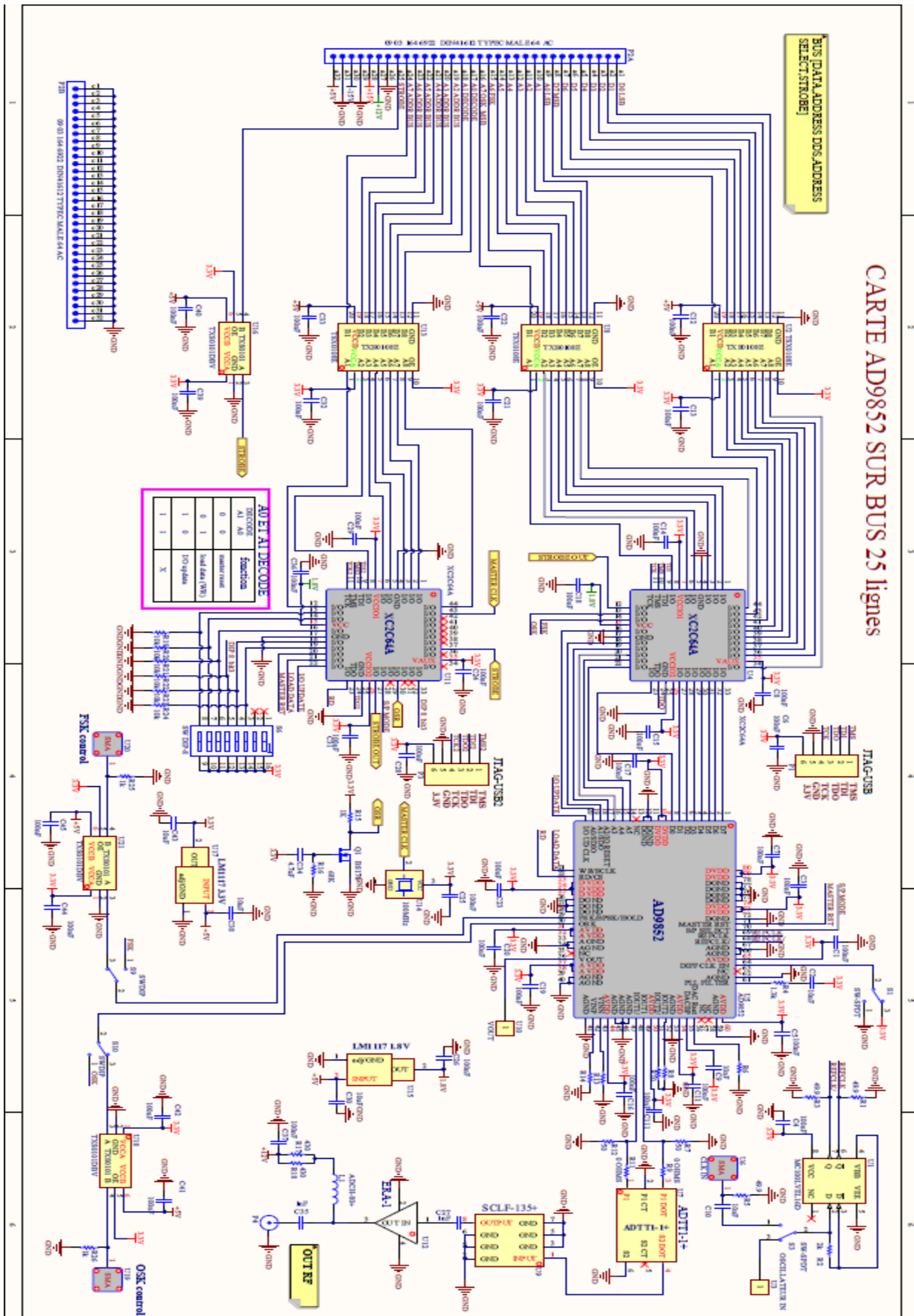
Table des registres: en rouge le mot de fréquence et en bleu mot d'amplitude.

AD9852

Table 9. Register Layout¹

Parallel Address (Hex)	Serial Address (Hex)	AD9852 Register Layout								Default Value (Hex)	
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
00	0	Phase Adjust Register 1 <13:8> (Bits 15, 14 don't care)					Phase 1				00
01		Phase Adjust Register 1 <7:0>									00
02	1	Phase Adjust Register 2 <13:8> (Bits 15, 14 don't care)			Phase Adjust Register 2 <7:0>			Phase 2			00
03		Phase Adjust Register 2 <13:8> (Bits 15, 14 don't care)			Phase Adjust Register 2 <7:0>			Phase 2			00
04	2	Frequency tuning word <47:40>					Frequency 1				00
05		Frequency Tuning Word 1 <39:32>									00
06		Frequency Tuning Word 1 <31:24>									00
07		Frequency Tuning Word 1 <23:16>									00
08		Frequency Tuning Word 1 <15:8>									00
09		Frequency Tuning Word 1 <7:0>									00
0A	3	Frequency Tuning Word 2 <47:40>					Frequency 2				00
0B		Frequency Tuning Word 2 <39:32>									00
0C		Frequency Tuning Word 2 <31:24>									00
0D		Frequency Tuning Word 2 <23:16>									00
0E		Frequency Tuning Word 2 <15:8>									00
0F		Frequency Tuning Word 2 <7:0>									00
10		Delta frequency word <47:40>									00
11		Delta frequency word <39:32>									00
12		Delta frequency word <31:24>									00
13		Delta frequency word <23:16>									00
14		Delta frequency word <15:8>									00
15		Delta frequency word <7:0>									00
16	5	Update clock <31:24>									00
17		Update clock <23:16>									00
18		Update clock <15:8>									00
19		Update clock <7:0>									00
1A	6	Ramp rate clock <19:16> (Bits 23, 22, 21, 20, don't care)									00
1B		Ramp rate clock <15:8>									00
1C		Ramp rate clock <7:0>									00
1D	7	Don't care	Don't care	Don't care	Comp PD	Reserved, always low	Control DAC PD	DAC PD	DIG PD	10	
1E		Don't care	PLL range	Bypass PLL	Ref Mult 4	Ref Mult 3	Ref Mult 2	Ref Mult 1	Ref Mult 0	64	
1F		CLR ACC1	CLR ACC2	Triangle	Don't care	Mode 2	Mode 1	Mode 0	Int/Ext update clock	01	
20		Don't care	Bypass inv sinc	Amplitude word			Don't care	LSB first	SDO active CR [0]	20	
21	8	Output shaped keying multiplier <11:8> (Bits 15, 14, 13, 12 don't care)									00
22		Output shaped keying multiplier <7:0>									00
23	9	Don't care									00
24		Don't care									00
25	A	Output shaped keying ramp rate <7:0>									80
26	B	Control DAC <11:8> (Bits 15, 14, 13, 12 don't care)									00
27		Control DAC <7:0> (Data is required to be in twos complement format)									00

3. Le schéma électronique et le PCB de la carte ad9852



PCB BOARD 3D OF AD9852

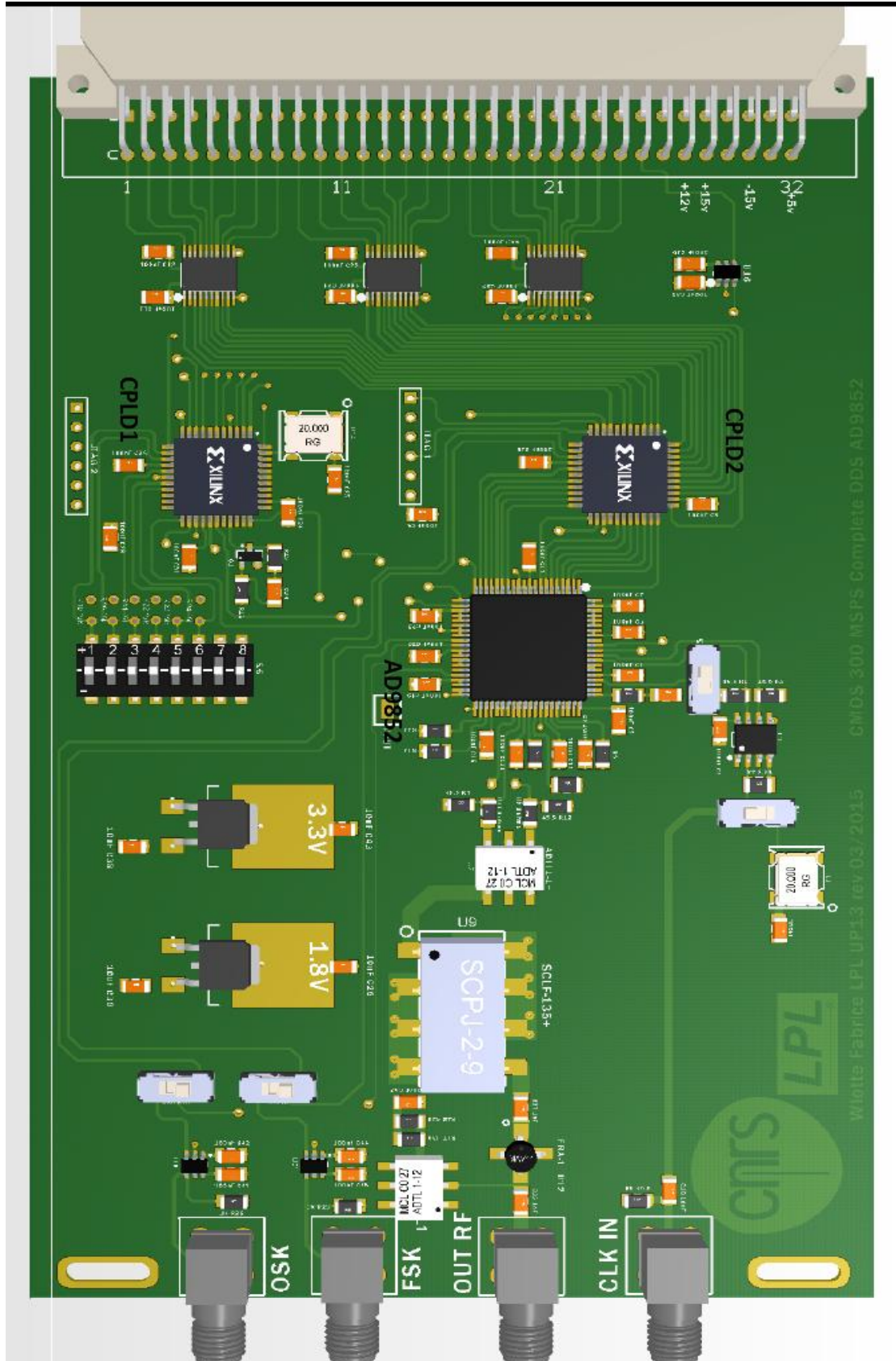
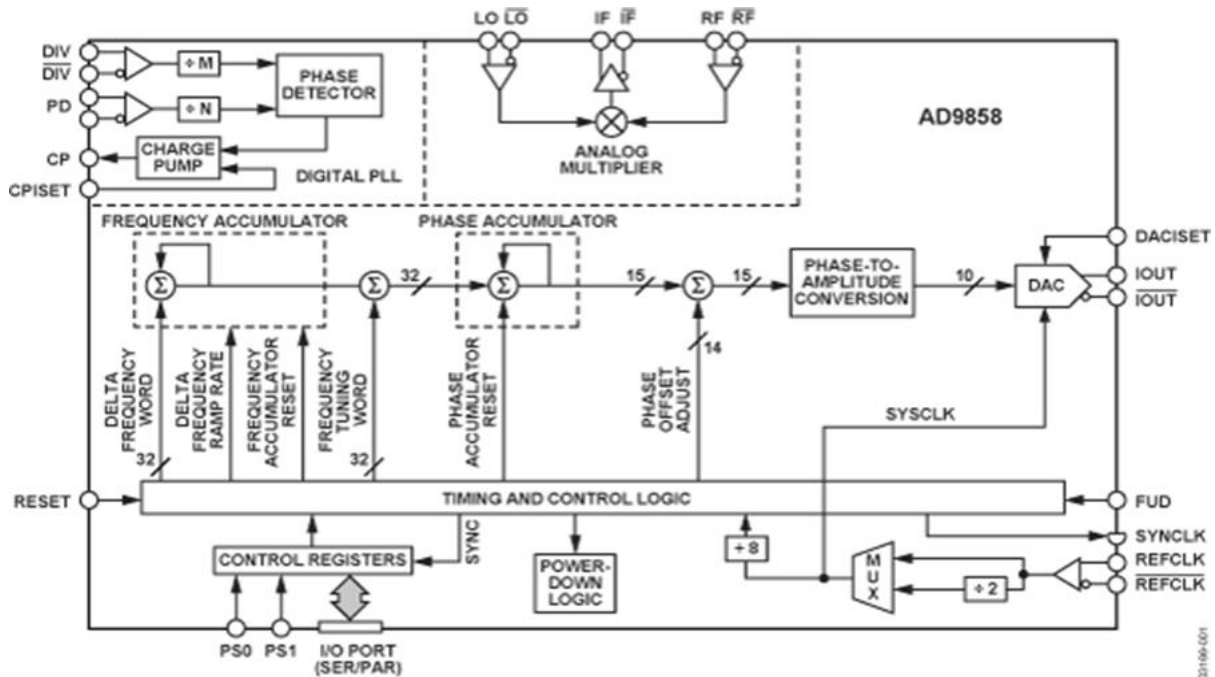


Diagramme fonctionnel de l'AD9858: L'AD9858



Principales fonctions:

REFERENCE CLOCK IN 1GHz si bit6=1 registre CFR ou 2GHz si bit6 =0
MASTER RESET bit actif sur front logic 1 initialise le DDS et met la sortie RF à zéro.
IOUT IOUT/ est la sortie RF principale 0dBm en général.
FUD est le bit de chargement des données dans le DDS ou I/O Update.
PS0/PS1 bits actif sur front logic 1 active les 4 profils de fréquences à précharger.
8-BIT PARALLEL LOAD bus 8bits pour chargement des données actif si mode parallèle sélectionné. D0-D7
6-BIT ADDRESS bus 6bits pour chargement des adresses en mode parallèle ADDR0 à ADDR5.
SERIAL/PARALLEL SELECT (SP SELECT) bit de sélection des modes de programmation du DDS: = 1 si chargement parallèle, 0 si chargement série (pin 91 du DDS).
DIFF/SINGLE SELECT bit de selection single ended or differential clock input. pin 64 du DDS.
WRITE OR WR/SCLK si mode Parallèle => bit de chargement I/O port buffers, actif sur front descendant. pin 19 du DDS.
RD/CS/ :active low pin 22 du DDS.
CP PD DIV DIV/ LO LO/ IF IF/ RF RF/ Non utilisé pour cette application.

Table des registres: en rouge le mot de fréquence et en bleu mot d'amplitude

L'amplitude n'a pas de registre, l'AD9858 est par défaut Full Scale.

AD9858

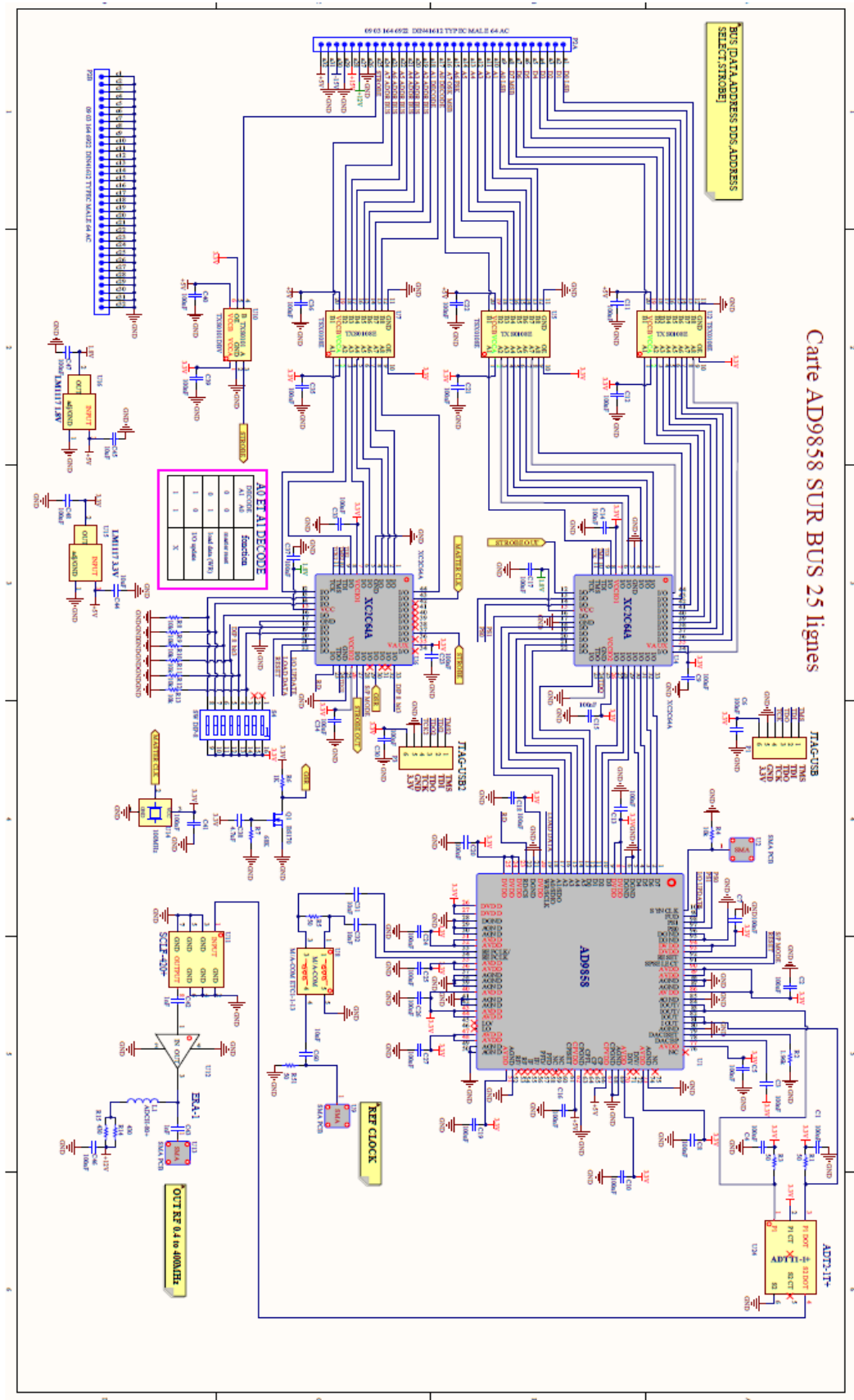
REGISTER MAP

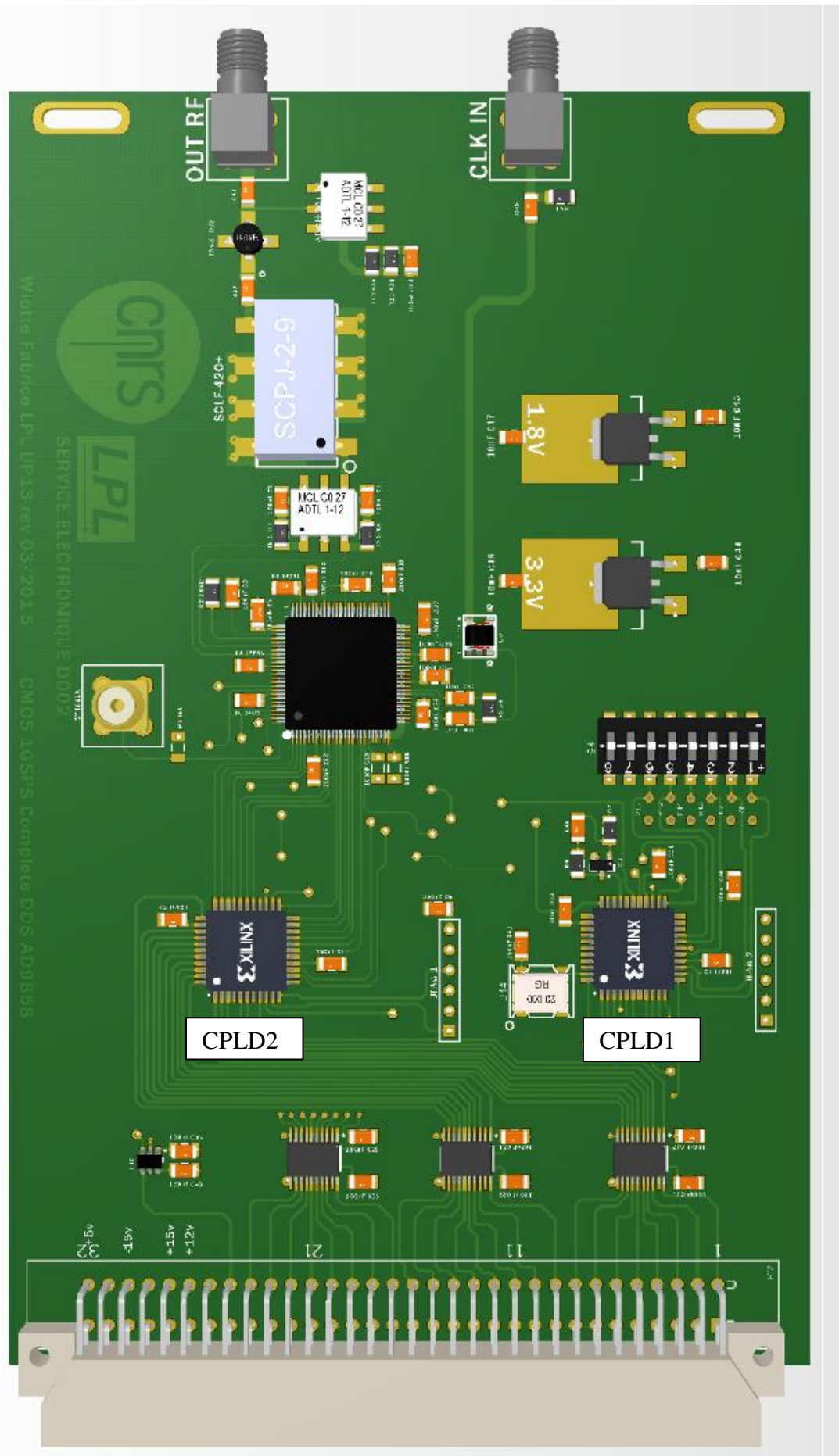
The registers are listed in Table 6. The serial address and parallel address numbers associated with each of the registers are shown in hexadecimal format. Square brackets [] are used to reference specific bits or ranges of bits. For example, [3] designates Bit 3, and [7:3] designates the range of bits from 7 down to 3, inclusive.

Table 6.

Register Name	Address		(MSB)							(LSB)	Default Value	Profile	
	Ser	Par	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0			
Control function register (CFR)	0x00	0x00 [7:0]	Not used	2 GHz divider disable	SYNCLK disable	Mixer power-down	Phase detect power-down	Power-down	SDIO input only	LSB first	0x18	N/A	
		0x01 [15:8]	Freq. sweep enable	Enable sine output	Charge pump offset	Phase detector divider ratio (N) (see Table 10)	Charge pump polarity	Phase detector divider ratio (M) (see Table 11)			0x00	N/A	
		0x02 [23:16]	Auto Clr freq. accum	Auto Clr phase accum	Load delta freq timer	Clear freq accum	Clear phase accum	Not used	Fast lock enable	FTW for fast lock	0x00	N/A	
		0x03 [31:24]	Frequency detect mode charge pump current (see Table 7)		Final closed-loop mode charge pump current (see Table 8)			Wide closed-loop mode charge pump current (see Table 9)			0x00	N/A	
Delta freq. tuning word (DFTW)	0x01	0x04	Delta Frequency Word[7:0]									N/A	
		0x05	Delta Frequency Word[15:8]									N/A	
		0x06	Delta Frequency Word[23:16]									N/A	
		0x07	Delta Frequency Word[31:24]									N/A	
Delta frequency ramp rate (DFRRW)	0x02	0x08	Delta Frequency Ramp Rate Word[7:0]									N/A	
		0x09	Delta Frequency Ramp Rate Word[15:8] fréquence tuning word									N/A	
Frequency Tuning Word 0 (FTW0)	0x03	0x0A	Frequency Tuning Word 0[7:0]									0x00	0
		0x0B	Frequency Tuning Word 0[15:8]									0x00	0
		0x0C	Frequency Tuning Word 0[23:16]									0x00	0
		0x0D	Frequency Tuning Word 0[31:24]									0x00	0
Phase Offset Word 0 (POW0)	0x04	0x0E	Phase Offset Word 0[7:0]									0x00	0
		0x0F	Not used	Phase Offset Word 0[13:8]								0x00	0
Frequency Tuning Word 1 (FTW1)	0x05	0x10	Frequency Tuning Word 1[7:0]										1
		0x11	Frequency Tuning Word 1[15:8]										1
		0x12	Frequency Tuning Word 1[23:16]										1
		0x13	Frequency Tuning Word 1[31:24]										1
Phase Offset Word 1 (POW1)	0x06	0x14	Phase Offset Word 1[7:0]										1
		0x15	Not used	Phase Offset Word 1[13:8]									1
Frequency Tuning Word 2 (FTW2)	0x07	0x16	Frequency Tuning Word 2[7:0]										2
		0x17	Frequency Tuning Word 2[15:8]										2
		0x18	Frequency Tuning Word 2[23:16]										2
		0x19	Frequency Tuning Word 2[31:24]										2

5. Le schéma électronique et le PCB de la carte AD9858





6. Présentation des composants logiques programmables (XC2C64A).

CPLD XC2C64A

Pour initialiser les cartes DDS et transférer les données aux DDS via le bus nous utilisons des composants programmables du type CPLD (circuit logique programmable) associés à des translateurs de niveaux pour passer du niveau TTL (5v) au niveau LVCMOS (3.3v).

Le composant utilisé est un CPLD de la famille XILINX XC2C64A 64 cellules.

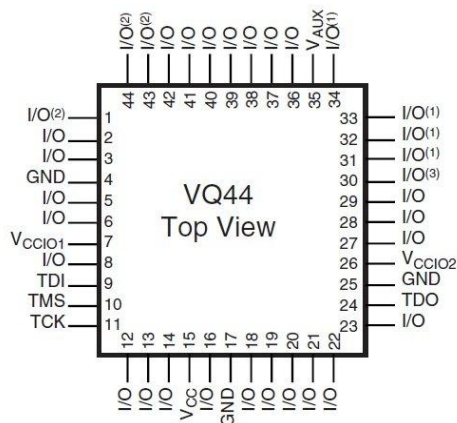
Ce composant initialise la carte DDS, assure le décodage d'adresses d'appareils, et surtout la synchronisation des pins du DDS (MASTER RESET, LOAD DATA et I/O UPDATE).

Les cartes DDS dispose de deux CPLD le premier en bas (CPLD1) de la carte pour la synchronisation et l'initialisation du DDS, le deuxième (CPLD2) recopie les données en entrée du bus après autorisation du CPLD1.

La programmation des CPLD se fait grâce à l'environnement ISE design suite 14 ou ultérieur téléchargeable gratuitement sur le site de XILINX (ISE WEB PACK) ;

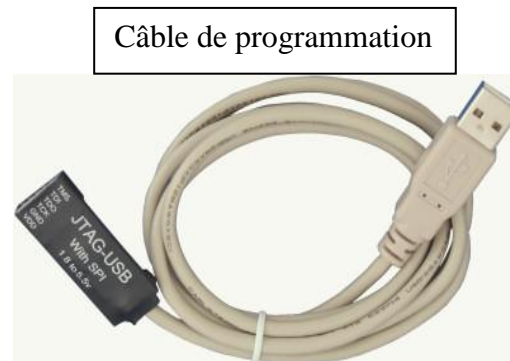
Ci-joint dans ce document les codes sources pour charger les deux CPLD. Un câble JTAG-USB est indispensable pour la programmation.

Package Pinout Diagrams



- (1) - Global Output Enable
- (2) - Global Clock
- (3) - Global Set/Reset

Figure 6: VQ44 Package



Environnement de programmation XILINX ISE WEB PACK à télécharger gratuitement sur <http://www.xilinx.com/products/design-tools/ise-design-suite/ise-webpack.html>

The screenshot shows the Xilinx ISE WebPack IDE interface. The main window displays the VHDL code for the entity 'addressbus_and_decode'. The code includes library declarations for IEEE, generic parameters for 'n', and a behavioral architecture. The architecture defines signals for internal bus, strobe, and selection lines, and uses a process to handle the master clock and reset signals.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.STD_LOGIC_ARITH.all;
4 use IEEE.STD_LOGIC_UNSIGNED.all;
5
6
7 entity addressbus_and_decode is
8     GENERIC (n: INTEGER := 20);
9     Port (
10         ADDR_BUS_in : in  STD_LOGIC_VECTOR (5 downto 0);
11         A0_A1_address: in  STD_LOGIC_VECTOR (1 downto 0);
12         STROBE : in  STD_LOGIC;
13         MASTER_CLOCK : in  STD_LOGIC;
14         STROBE_OUT : out  STD_LOGIC;
15         IO_UPDATE : out  STD_LOGIC;
16         RD : out  STD_LOGIC;
17         LOAD_DATA : out  STD_LOGIC;
18         MASTER_RESET : out  STD_LOGIC;
19         SP_MODE : out  STD_LOGIC;
20         CONTROL_DIP8 : in  STD_LOGIC_VECTOR (5 downto 0)
21     );
22
23 end addressbus_and_decode;
24
25
26 architecture Behavioral of addressbus_and_decode is
27
28     SIGNAL internal: STD_LOGIC_VECTOR (n-1 DOWNTO 0);
29     SIGNAL STROBE_OUT_bis : STD_LOGIC;
30     SIGNAL SEL0 : STD_LOGIC;
31     SIGNAL SEL1 : STD_LOGIC;
32     SIGNAL SEL2 : STD_LOGIC;
33     SIGNAL SEL3 : STD_LOGIC;
34
35     begin
36
37     --registre & d&ecalage Tempo STROBE
38     PROCESS(MASTER_CLOCK, STROBE)
39     BEGIN
40     IF (STROBE = '0') THEN
41         internal <= (OTHERS => '0');
42     ELSIF (MASTER_CLOCK'EVENT AND MASTER_CLOCK='1') THEN
43         internal <= STROBE & internal(internal'LEFT_DOWNTO 1);
```

The screenshot shows the Xilinx ISE WebPack IDE interface displaying the Design Summary for the 'addressbus_and_decode.ucf' file. The summary lists various net assignments for the design, including address bus lines, control signals, and I/O pins.

```
1 //NET "IO_UPDATE" LOC = "P22" ;
2 #PACE: Start of Constraints generated by PACE
3
4 #PACE: Start of PACE I/O Pin Assignments
5 NET "A0_A1_address<0>" LOC = "P12" ;
6 NET "A0_A1_address<1>" LOC = "P8" ;
7 NET "ADDR_BUS_in<0>" LOC = "P6" ;
8 NET "ADDR_BUS_in<1>" LOC = "P5" ;
9 NET "ADDR_BUS_in<2>" LOC = "P3" ;
10 NET "ADDR_BUS_in<3>" LOC = "P2" ;
11 NET "ADDR_BUS_in<4>" LOC = "P1" ;
12 NET "ADDR_BUS_in<5>" LOC = "P4" ;
13 NET "CONTROL_DIP8<0>" LOC = "P13" ;
14 NET "CONTROL_DIP8<1>" LOC = "P14" ;
15 NET "CONTROL_DIP8<2>" LOC = "P16" ;
16 NET "CONTROL_DIP8<3>" LOC = "P33" ;
17 NET "CONTROL_DIP8<4>" LOC = "P18" ;
18 NET "CONTROL_DIP8<5>" LOC = "P19" ;
19 NET "LOAD_DATA" LOC = "P21" ;
20 NET "MASTER_CLOCK" LOC = "P43" ;
21 NET "MASTER_RESET" LOC = "P20" ;
22 NET "RD" LOC = "P23" ;
23 NET "SP_MODE" LOC = "P28" ;
24 NET "STROBE" LOC = "P37" ;
25 NET "STROBE_OUT" LOC = "P27" ;
26 NET "IO_UPDATE" LOC = "P22" ;
27 #PACE: Start of PACE Area Constraints
28
29 #PACE: Start of PACE Prohibit Constraints
30
31 #PACE: End of Constraints generated by PACE
32
```

7. Les programmes associés aux CPLD

VHDL pour CPLD1 AD9852

```
--programme pour CPLD1 carte DDS AD9852 et bus NI-

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity addressbus_and_decode is

    GENERIC (n: INTEGER := 20);
    Port (
        ADDR_BUS_in : in  STD_LOGIC_VECTOR (5 downto 0);
        A0_A1_address: in  STD_LOGIC_VECTOR (1 downto 0);
        STROBE      : in  STD_LOGIC;
        MASTER_CLOCK : in  STD_LOGIC;
        STROBE_OUT  : out STD_LOGIC;
        --IO_UPDATE : out STD_LOGIC; -- par défaut mode dds
        RD          : out STD_LOGIC;
        LOAD_DATA   : out STD_LOGIC;
        MASTER_RESET : out STD_LOGIC;
        SP_MODE     : out STD_LOGIC;
        CONTROL_DIP8 : in  STD_LOGIC_VECTOR (5 downto 0)
    );

end addressbus_and_decode;

architecture Behavioral of addressbus_and_decode is

    SIGNAL internal: STD_LOGIC_VECTOR (n-1 DOWNTO 0);
    SIGNAL STROBE_OUT_bis : STD_LOGIC;
    SIGNAL SEL0 : STD_LOGIC;
    SIGNAL SEL1 : STD_LOGIC;
    SIGNAL SEL2 : STD_LOGIC;
    SIGNAL SEL3 : STD_LOGIC;
    SIGNAL reset : STD_LOGIC;
    SIGNAL load : STD_LOGIC;

Begin

    SP_MODE <='1';
    RD <='1';

    --registre a décalage Tempo STROBE générateur de temporisation pour
    décalage
    PROCESS(MASTER_CLOCK, STROBE)
    BEGIN
        IF (STROBE ='0') THEN
            internal <= (OTHERS => '0');
        ELSIF (MASTER_CLOCK'EVENT AND MASTER_CLOCK='1') THEN
            internal <= STROBE & internal(internal'LEFT DOWNTO 1);
        END IF;
    END PROCESS;

end Behavioral;
```



```
-- 2 bits address A0 et A1 decoder 2 to 4 and control ADDR = CONTROL_dip8

process(A0_A1_address,ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(15) = '1' and ADDR_BUS_in = CONTROL_DIP8 then
case A0_A1_address is
when "00"=> SEL0 <='1'; SEL1<='0'; SEL2<='0'; SEL3<='0';
when "01"=> SEL0 <='0'; SEL1<='1'; SEL2<='0'; SEL3<='0';
when "10"=> SEL0 <='0'; SEL1<='0'; SEL2<='1'; SEL3<='0';
when "11"=> SEL0 <='0'; SEL1<='0'; SEL2<='0'; SEL3<='1';
when others => SEL0 <='Z'; SEL1<='Z'; SEL2 <='Z'; SEL3 <='Z';
end case;
else
SEL0 <='Z'; SEL1<='Z'; SEL2 <='Z'; SEL3 <='Z';
end if;
end process;

STROBE_OUT <= STROBE_OUT_bis;

--STROBE out pour autorisation sur CPLD2 latches décalage de 50ns /strobe
process(internal(15),ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(15) = '1' and ADDR_BUS_in = CONTROL_DIP8 and SEL0 = '0' then
STROBE_OUT_bis <='1';
else
STROBE_OUT_bis <='0';
end if;
end process;

-- control bit MASTER_RESET décalage de 50ns /strobe
process(internal(15),SEL0,ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(15) = '1' and SEL0 = '1' and ADDR_BUS_in = CONTROL_DIP8 then
MASTER_RESET <= '1';
else
MASTER_RESET <= '0';
end if;
end process;

-- control bit Load data into IO buffer(WR) décalage de 100ns /strobe
process(internal(10),SEL1,ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(10) = '1' and SEL1 = '1' and ADDR_BUS_in = CONTROL_DIP8 then
LOAD_DATA <= '0';
else
LOAD_DATA <= '1';
end if;
end process;
end Behavioral;
```

UCF implémentation des pins I/O CPLD1 AD9852 and AD9858

```
//NET "IO_UPDATE" LOC = "P22" ;
#PACE: Start of Constraints generated by PACE
#PACE: Start of PACE I/O Pin Assignments
NET "A0_A1_address<0>" LOC = "P12" ;
NET "A0_A1_address<1>" LOC = "P8" ;
NET "ADDR_BUS_in<0>" LOC = "P6" ;
NET "ADDR_BUS_in<1>" LOC = "P5" ;
NET "ADDR_BUS_in<2>" LOC = "P3" ;
NET "ADDR_BUS_in<3>" LOC = "P2" ;
NET "ADDR_BUS_in<4>" LOC = "P1" ;
NET "ADDR_BUS_in<5>" LOC = "P44" ;
NET "CONTROL_DIP8<0>" LOC = "P13" ;
NET "CONTROL_DIP8<1>" LOC = "P14" ;
NET "CONTROL_DIP8<2>" LOC = "P16" ;
NET "CONTROL_DIP8<3>" LOC = "P33" ;
NET "CONTROL_DIP8<4>" LOC = "P18" ;
NET "CONTROL_DIP8<5>" LOC = "P19" ;
NET "LOAD_DATA" LOC = "P21" ;
NET "MASTER_CLOCK" LOC = "P43" ;
NET "MASTER_RESET" LOC = "P20" ;
NET "RD" LOC = "P23" ;
NET "SP_MODE" LOC = "P28" ;
NET "STROBE" LOC = "P37" ;
NET "STROBE_OUT" LOC = "P27" ;
NET "IO_UPDATE" LOC = "P22" ;
#PACE: Start of PACE Area Constraints
#PACE: Start of PACE Prohibit Constraints
#PACE: End of Constraints generated by PACE
```

VHDL pour CPLD1 AD9858

--programme pour CPLD1 carte DDS AD9858 et bus NI-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity addressbus_and_decode is
  GENERIC (n: INTEGER := 20);
  Port (
    ADDR_BUS_in : in  STD_LOGIC_VECTOR (5 downto 0);
    A0_A1_address: in  STD_LOGIC_VECTOR (1 downto 0);
    STROBE      : in  STD_LOGIC;
    SYNCLK      : in  STD_LOGIC;
    MASTER_CLOCK : in  STD_LOGIC;
    STROBE_OUT  : out STD_LOGIC;
    IO_UPDATE   : out STD_LOGIC;
    RD          : out STD_LOGIC;
    LOAD_DATA   : out STD_LOGIC;
    MASTER_RESET : out STD_LOGIC;
    SP_MODE     : out STD_LOGIC;
    CONTROL_DIP8 : in  STD_LOGIC_VECTOR (5 downto 0)
  );
end addressbus_and_decode;

architecture Behavioral of addressbus_and_decode is

  SIGNAL internal: STD_LOGIC_VECTOR (n-1 DOWNT0 0);
  SIGNAL STROBE_OUT_bis : STD_LOGIC;
  SIGNAL SEL0 : STD_LOGIC;
  SIGNAL SEL1 : STD_LOGIC;
  SIGNAL SEL2 : STD_LOGIC;
  SIGNAL SEL3 : STD_LOGIC;
  SIGNAL reset : STD_LOGIC;
  SIGNAL load : STD_LOGIC;
  signal count_100Hz : STD_LOGIC_VECTOR(1 DOWNT0 0);
  signal clock_100Hz_int: STD_LOGIC;
  signal FUD : STD_LOGIC_VECTOR(9 DOWNT0 0);
  signal FUD_int : STD_LOGIC;

begin

  --registre a décalage Tempo STROBE générateur de temporisation pour
  décalage
  PROCESS(MASTER_CLOCK, STROBE)
  BEGIN
    IF (STROBE ='0') THEN
      internal <= (OTHERS => '0');
    ELSIF (MASTER_CLOCK'EVENT AND MASTER_CLOCK='1') THEN
      internal <= STROBE & internal(internal'LEFT DOWNT0 1);
    END IF;
  END PROCESS;

end Behavioral;
```

```
-- 2 bits address A0 et A1 decoder 2 to 4 and control ADDR = CONTROL_dip8
process(A0_A1_address,ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(15) ='1' and ADDR_BUS_in = CONTROL_DIP8 then
case A0_A1_address is
when "00"=> SEL0 <='1'; SEL1<='0'; SEL2<='0'; SEL3<='0';
when "01"=> SEL0 <='0'; SEL1<='1'; SEL2<='0'; SEL3<='0';
when "10"=> SEL0 <='0'; SEL1<='0'; SEL2<='1'; SEL3<='0';
when "11"=> SEL0 <='0'; SEL1<='0'; SEL2<='0'; SEL3<='1';
when others => SEL0 <='Z'; SEL1<='Z'; SEL2 <='Z'; SEL3 <='Z';
end case;
end if;
end process;

STROBE_OUT <= STROBE_OUT_bis;
IO_UPDATE <= STROBE;
SP_MODE <='1';
RD <='1';

--STROBE out pour autorisation sur CPLD2 latches décalage de 50ns /strobe
process(internal(15),ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(15) ='1' and ADDR_BUS_in = CONTROL_DIP8 and SEL0 ='0' then
STROBE_OUT_bis <='1';
else
STROBE_OUT_bis <='0';
end if;
end process;

-- control bit MASTER_RESET décalage de 50ns /strobe
process(internal(15),SEL0,ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(15) = '1' and SEL0 ='1' and ADDR_BUS_in = CONTROL_DIP8 then
MASTER_RESET <= '1';
else
MASTER_RESET <= '0';
end if;
end process;

-- control bit Load data into IO buffer(WR) décalage de 100ns /strobe
process(internal(10),SEL1,ADDR_BUS_in,CONTROL_DIP8)
begin
if internal(10)= '1' and SEL1 ='1' and ADDR_BUS_in = CONTROL_DIP8 then
LOAD_DATA <= '0';
else
LOAD_DATA <= '1';
end if;
end process;
end Behavioral;
```

VHDL pour CPLD2 AD9852 et AD9858

--Programme CPLD2 pour AD9852 et AD9858--

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_cpld_haut is

    Port (
        DATA_in : in  STD_LOGIC_VECTOR (7 downto 0);
        ADDRESS_in : in  STD_LOGIC_VECTOR (7 downto 0);
        DATA_out : out STD_LOGIC_VECTOR (7 downto 0);
        ADDRESS_out : out STD_LOGIC_VECTOR (7 downto 0);
        STROBE_in : in  STD_LOGIC
    );

end test_cpld_haut;

architecture Behavioral of test_cpld_haut is

begin

    -- octal 2 d-type latches 3 états, 8bits
    process(STROBE_in,DATA_in,ADDRESS_in)
    begin
        if STROBE_in ='1' then
            DATA_out <= DATA_in;
            ADDRESS_out <= ADDRESS_in;
        end if;
    end process;

end Behavioral;
```

UCF implémentation des pins I/O CPLD2 AD9852 et AD9858

```
NET "ADDRESS_in<0>" LOC = "P42" ;
NET "ADDRESS_in<1>" LOC = "P41" ;
NET "ADDRESS_in<2>" LOC = "P40" ;
NET "ADDRESS_in<3>" LOC = "P39" ;
NET "ADDRESS_in<4>" LOC = "P38" ;
NET "ADDRESS_in<5>" LOC = "P37" ;
NET "ADDRESS_in<6>" LOC = "P36" ;
NET "ADDRESS_in<7>" LOC = "P34" ;

NET "DATA_in<0>" LOC = "P8" ;
NET "DATA_in<1>" LOC = "P6" ;
NET "DATA_in<2>" LOC = "P5" ;
NET "DATA_in<3>" LOC = "P3" ;
NET "DATA_in<4>" LOC = "P2" ;
NET "DATA_in<5>" LOC = "P1" ;
NET "DATA_in<6>" LOC = "P44" ;
NET "DATA_in<7>" LOC = "P43" ;
#PACE: Start of Constraints generated by PACE

#PACE: Start of PACE I/O Pin Assignments
NET "ADDRESS_out<0>" LOC = "P16" ;
NET "ADDRESS_out<1>" LOC = "P18" ;
NET "ADDRESS_out<2>" LOC = "P19" ;
NET "ADDRESS_out<3>" LOC = "P20" ;
NET "ADDRESS_out<4>" LOC = "P21" ;
NET "ADDRESS_out<5>" LOC = "P22" ;
NET "ADDRESS_out<6>" LOC = "P14" ;
NET "ADDRESS_out<7>" LOC = "P13" ;

NET "DATA_out<0>" LOC = "P23" ;
NET "DATA_out<1>" LOC = "P27" ;
NET "DATA_out<2>" LOC = "P28" ;
NET "DATA_out<3>" LOC = "P29" ;
NET "DATA_out<4>" LOC = "P30" ;
NET "DATA_out<5>" LOC = "P31" ;
NET "DATA_out<6>" LOC = "P32" ;
NET "DATA_out<7>" LOC = "P33" ;
//NET "PS0_PS1<0>" LOC = "P13" ;
//NET "PS0_PS1<1>" LOC = "P14" ;
NET "STROBE_in" LOC = "P12" ;

#PACE: Start of PACE Area Constraints

#PACE: Start of PACE Prohibit Constraints
```


8. Liste des composants carte DDS AD9852

Comment	Description	Designator	Footprint	Quantity
	LISTE MATERIEL AD9852	LISTE MATERIEL AD9852		
0 OHMS	Resistor	R9, R11	CL206	2
LM1117 1.8V	REGULATEUR CMS	U15	D2PAK_M	1
LM1117 3.3V	REGULATEUR CMS	U17	D2PAK_M	1
10uF 50V	Capacitor	C43, C38, C26, C30	CL206	4
REN	Resistor	R13, R14	CL206	0
1.91k	Resistor	R6	CL206	1
49.9	Resistor	R1, R3, R5, R8, R10, R7, R12	CL206	4
AD9852	dds ad9852	U5	LQFP80_M	1
ADCH-80+	Inductor	L1	ADCH-80+	1
ADTT1-1+	RF TRANSFORMER	U7	ADTT1-1+	1
10nF 50V	Capacitor	C2, C9, 10	CL206	3
4.7uF 50V	Capacitor	C34	CL206	1
1nF 50V	Capacitor	C27, C35	CL206	2
CLK IN	POINT TEST SMA	CLK IN	EMBASE SMA	1
ERA-1	amppli 20dB	U12	ERA-5	1
10k	Resistor	R19, R20, R21, R22, R23, R24		6
1k	Resistor	R26, R25, R15		3
68k	Resistor	R16	CL206	1
2.2k	Resistor	R2	CL206	1
JTAG-USB	Header, 6-Pin	JTAG 1	HDR1X6	1
JTAG-USB2	Header, 6-Pin	JTAG 2	HDR1X6	1
BS170	N-Channel MOSFET	Q1	SOT23_N	1
1.3k	Resistor	R4	CL206	1
430	Resistor	R17, R18	CL206	2
SW-SPDT	Angle Mounting, Vertical Actuation	S1, S3, S9, S10	HDR1X3	4
SW DIP-8	SWITCH 8 VOIES	S6	N-16	1
SCLF-135+	to 190MHz	U9	SCLF-190+	1
MC100VEL16D	Differential Receiver	U1	751-05_N	1
TSX0108E	Differential Receiver	U2, U8, U13	TSSOP20_M	3
XCC2G64A	TRANSLATOR	U4, U11	VQ64_N	2
XCC2G64A	CoolRunner-II CPLD	C16, C17, C18, C19, C20, C21, C22, C23, C24, C28, C31, C32, C33, C36, C37, C39, C40, C41, C42, C44, C45, C111	CL206	33
100nF	Capacitor	U3	QUARTZ 3D	1
QUARTZ	20MHz	U16, U21, U18	SOT23-6_N	3
TXS0101DBV	TRANSLATEUR NIVEAU			

Liste des composants carte DDS AD9858

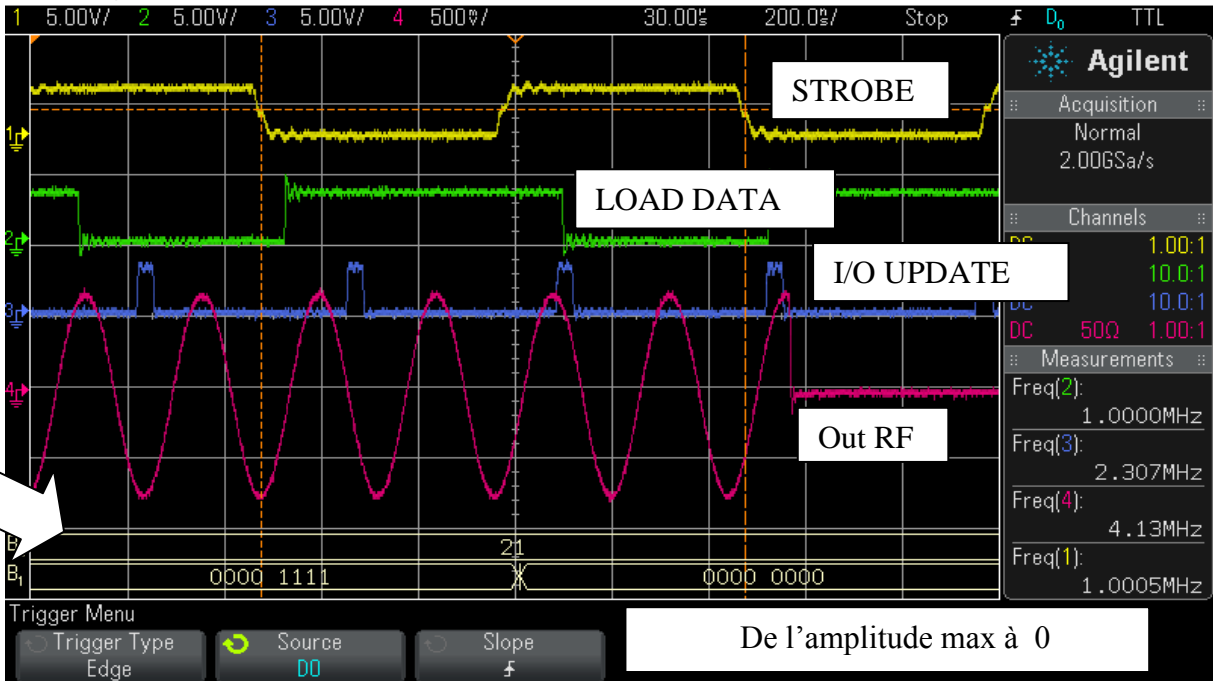
Comment	Description	Designator	Footprint	Quantity
AD9858	DDS	U1	TQ100_N	1
100nF-50V	Capacitor	C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26	C1206	37
10uF-50V	Capacitor	C44, C45, C47, C48	C1206	4
10nF-50V	Capacitor	C31, C32, C60	C1206	3
4.7uF-50V	Capacitor	C38	C1206	1
1nF-50V	Capacitor	C42, C43	C1206	4
ADCH-80+	Inductor	L1	ADTL-5-1	1
DINA41612	Eurocard DIN 41612, Type C, 64 Position	P2	0903164K922	1
JTAG-USB2	Header, 6-Pin	P1, P3	HDR1X6	2
BS170	N-Channel MOSFET	Q1	SOT23_N	1
50	Resistor	R1, R3, R5	C1206	3
1.96k	Resistor	R2	C1206	1
10k	Resistor	R4, R8, R9, R10, R11, R12, R13	C1206	7
68k	Resistor	R7	C1206	1
1k	Resistor	R6	C1206	1
430	Resistor	R14, R15	C1206	2
SW-DIP-8	DIP Switch, 8 Position, SPST	S4	HDR2X8	1
ADT2-1T+	RF TRANSFORMER	U24	ADT2-1T+	1
TSX0108E	8-BIT BIDIRECTIONAL VOLTAGE-LEVEL TRANSLATOR	U2, U5, U7	TSSOP20_M	3
SMA PCB	POINT TEST SMA	U10, U9, U13	EMBASE SMA CI	3
XC2C64A	XC2C64A CoolRunner-II CPLD	U4, U6	VQ644_N	2
ETCL-1-13	Transformateurs audio et de signaux 4.5-3000MHz	U8	ETCL-1-13TR	1
TXS0101DBV	TRANSLATEUR / DECALLAGE DE NIVEAU	U10	SOT23-6_N	1
SCLF-420+	Surface Mount Low pass Filter	U11	SCLF-420+	1
ERA-1	ampli 20dB	U12	ERA-5	1
100MHz	QUARTZ OSCILLATOR	U14	CLK_100MHz	1
LM1117 3.3V	REGULATEUR CMS	U15	D2PAK_M	1
LM1117 1.8V	REGULATEUR CMS	U16	D2PAK_M	1

Quelques mesures des signaux sur le BUS de données visualisation de la sortie RF

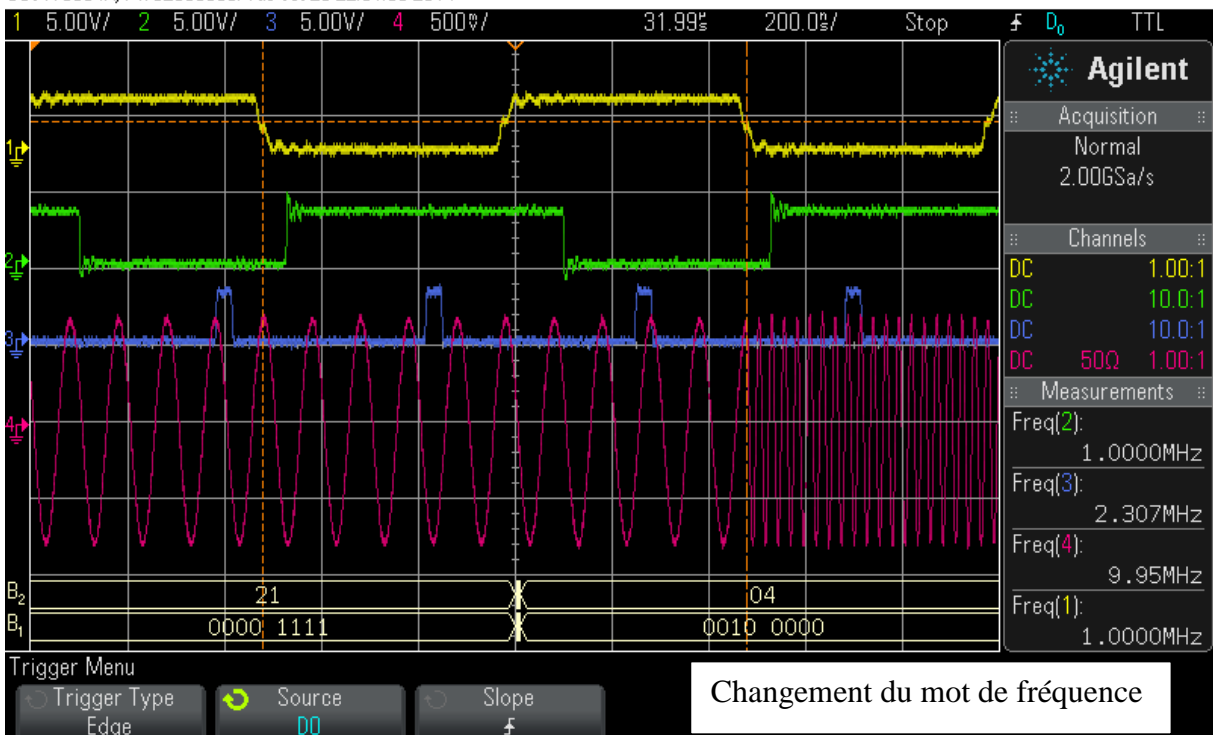
9. Mesures de signaux sur le bus de données de l'AD9852 :

- En ROUGE sortie RF d'un DDS AD9852
- En JAUNE : synchro STROBE @1MHz
- En VERT : LOAD DATA
- En BLEU : I/O UPDATE mode INTERNE (générer par le DDS)

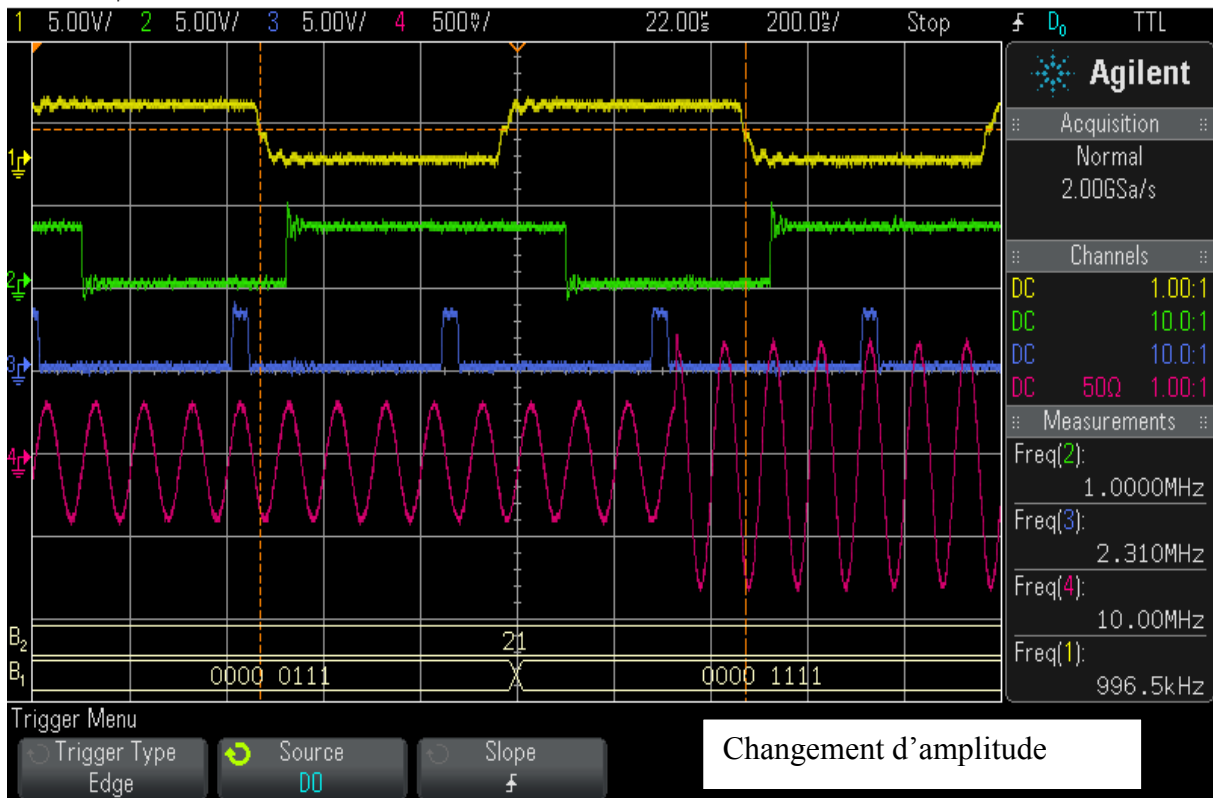
DSO-X 3054A, MY52390838: Tue Oct 28 22:01:11 2014



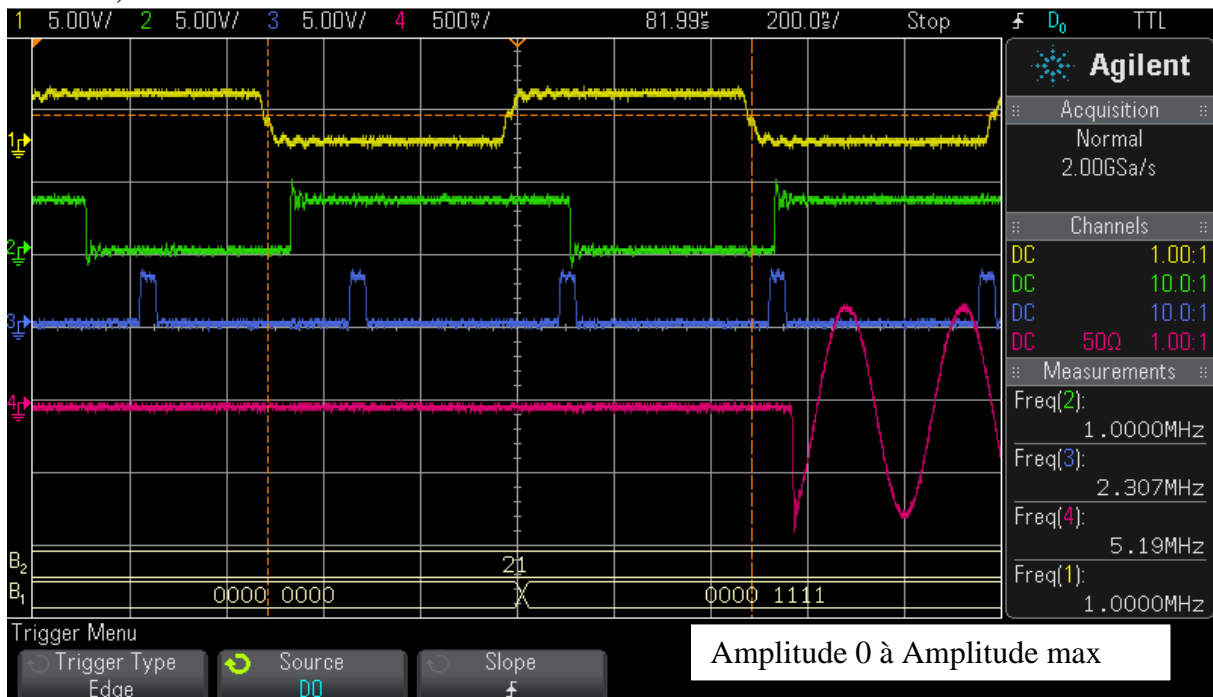
DSO-X 3054A, MY52390838: Tue Oct 28 22:01:38 2014



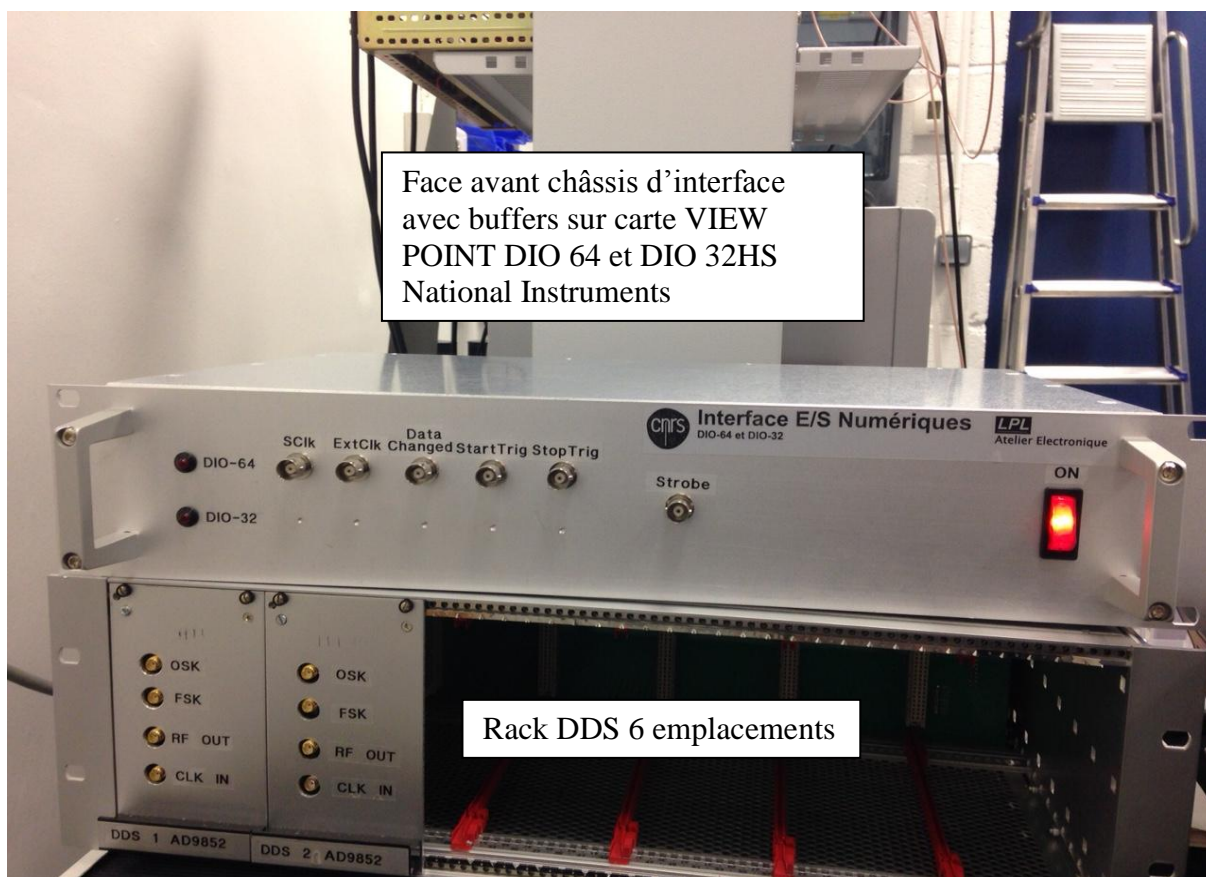
DSO-X 3054A, MY52390838: Tue Oct 28 22:02:53 2014



DSO-X 3054A, MY52390838: Tue Oct 28 22:03:51 2014



Vue assemblage des cartes électroniques



Face arrière châssis d'interface

