Composant PLL

J.-P. Cachemiche (CPPM)
Paul Bibron, Julien Langouët, Renaud Le Gac, Frédéric Réthoré

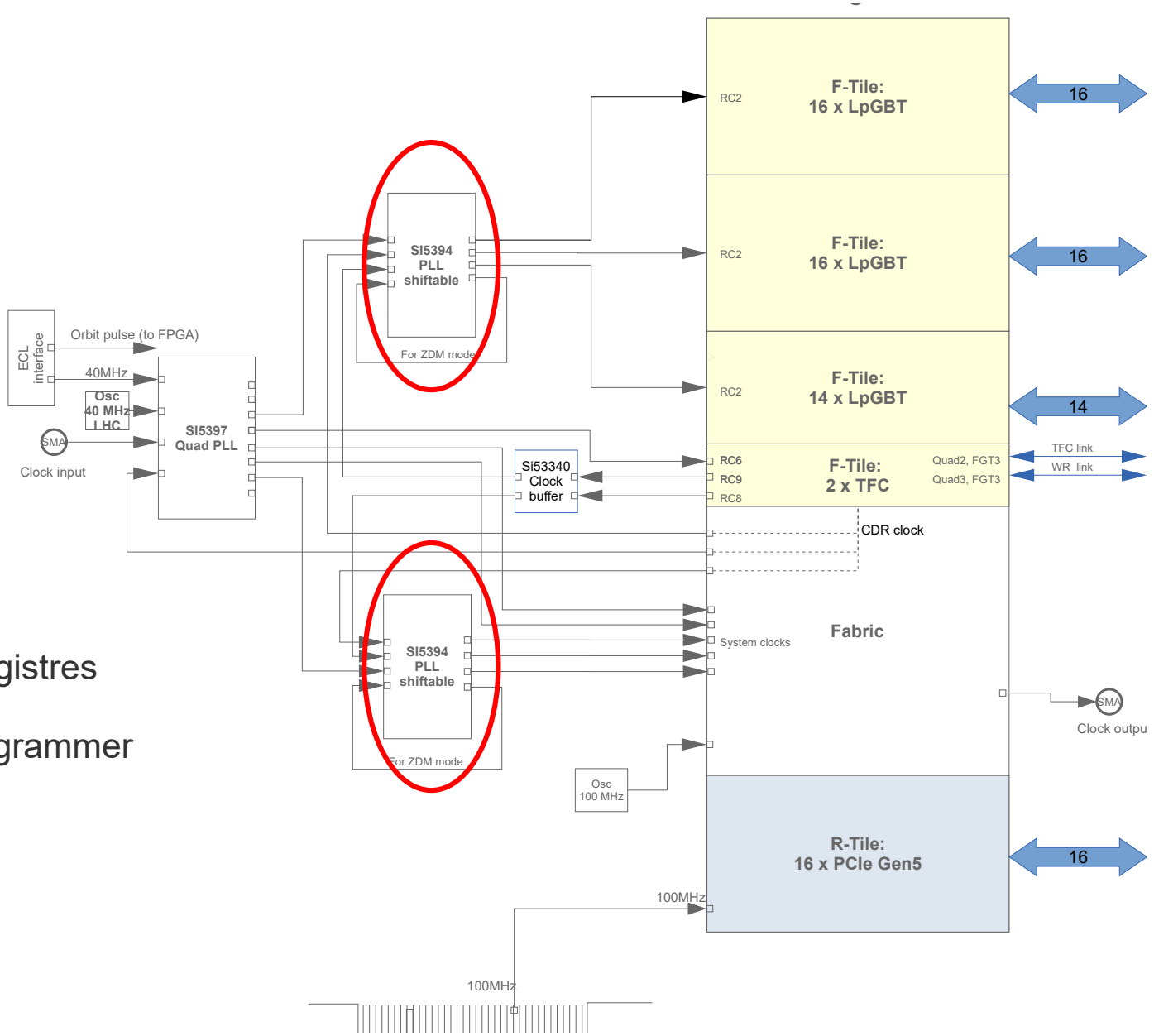
Arbre d’horloge

Utilisation de deux PLLs

- SI5394J
- SI5397J

Objet de l’exercice

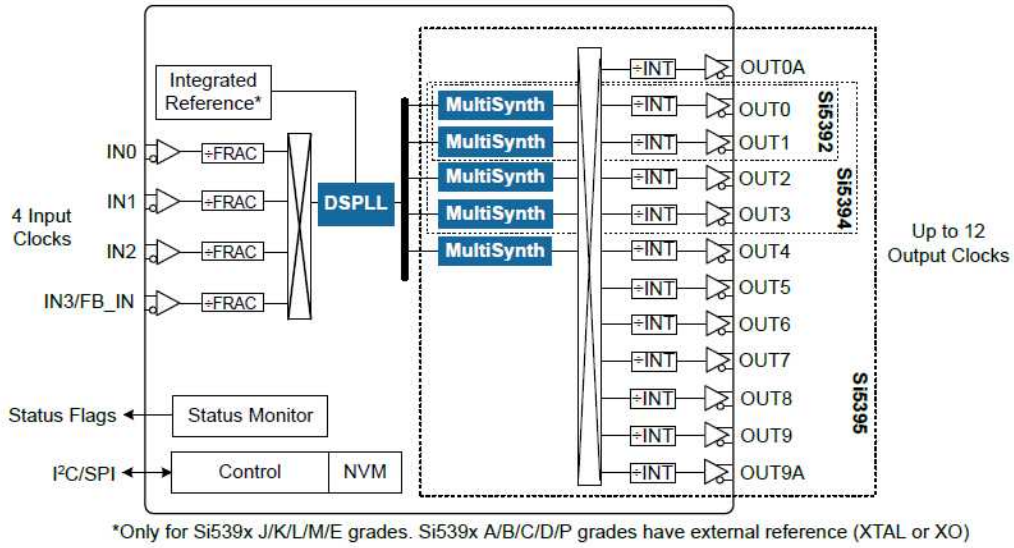
- Interfacer quelques registres de la PLL SI5394J
- Eventuellement la programmer



Description de la PLL SI5394J

Caractéristiques

- Ultra low phase jitter:
  - 85 fs RMS (integer mode)
  - 100 fs RMS (fractional mode)
- 1 entrée active parmi 4
- Input frequency range
  - Differential: 8 kHz to 750 MHz
  - LVCMOS: 8 kHz to 250 MHz
- 4 sorties
- Output frequency range
  - Differential: 100 Hz to 1028 MHz
  - LVCMOS: 100 Hz to 250 MHz
- Oscillateur interne intégré



Programmation

Registres répartis sur 9 pages

- Permet d'adresser plus de registres que ce que ne permet l'I2C
- Chaque page donne accès à 256 registres

Page	Start Address (Hex)	Start Address (Decimal)	Contents
Page 0	0000h	0	Alarms, interrupts, reset, other configuration
Page 1	0100h	256	Clock output configuration
Page 2	0200h	512	P,R dividers, scratch area
Page 3	0300h	768	Output N dividers, N divider Finc/Fdec
Page 4	0400h	1024	ZD mode configuration
Page 5	0500h	1280	M divider, BW, holdover, input switch, FINC/DEC
Page 9	0900h	2304	Control IO configuration
Page A	0A00h	2560	Multisynth Controls
Page B	0B00h	2816	Various
Page C	0C00h	3072	Various

- L'adresse 0x0001 de chaque page pointe vers un registre unique qui contient le numéro de page

Reg Address	Bit Field	Type	Name	Description
0x0001	7:0	R/W	PAGE	Selects one of 256 possible pages.

On every page, there is a "Page Register" located at address 0x01. When read, it indicates the current page. When written, it changes the page to the value entered. There are page registers at addresses 0x0001, 0x0101, 0x0201, 0x0301, … etc.

But de l'exercice

Ecrire un objet avec une classe permettant d'accéder aux registres d'identification du chip

- Part number
- Device grade
- Revision

Table 16.2. 0x0002–0x0003 Base Part Number

Reg Address	Bit Field	Type	Name	Value	Description
0x0002	7:0	R	PN_BASE	0x95	Four-digit "base" part number, one nibble per digit.
0x0003	15:8	R	PN_BASE	0x53	Example: Si5395A-A-GM. The base part number (OPN) is 5395, which is stored in this register.

Refer to the device data sheet Ordering Guide section for more information about device grades.

Table 16.3. 0x0004 Device Grade

Reg Address	Bit Field	Type	Name	Description
0x0004	7:0	R	GRADE	One ASCII character indicating the device speed/synthesis mode. 0 = A, 1 = B, 2 = C, 3 = D , 4 =E 9=J, 10=K,11=L,12=M,15=P etc

Table 16.4. 0x0005 Device Revision

Reg Address	Bit Field	Type	Name	Description
0x0005	7:0	R	DEVICE_REV	One ASCII character indicating the device revision level. 0 = A; 1 = B, etc. Example Si5395C-A12345-GM, the device revision is "A" and stored as 0.

Puis afficher les information avec un programme appelant cet objet

Supplément

Si le temps le permet, programmer a PLL

Principe

- On utilise le wizard pour générer la configuration
- On exporte le jeu de registres à programmer sous forme d'un fichier texte
- On extrait les champs du fichier texte avec Python pour programmer les registres

```
# Si534x/7x/8x/9x Registers Script
#
# Part: Si5394
# Project File: <not saved>
# Design ID: <none>
# Includes Pre/Post Download Control Register Writes: Yes
# Die Revision: C0
# Creator: ClockBuilder Pro v4.4 [2022-03-30]
# Created On: 2022-06-06 18:09:09 GMT+02:00
Address,Data
#
# Start configuration preamble
0x0B24,0xC0
0x0B25,0x00
0x0540,0x01
# End configuration preamble
#
# Delay 300 msec
#    Delay is worst case time for device to complete any calibration
#    that is running due to device state change previous to this script
#    being processed.
#
# Start configuration registers
0x0006,0x00
0x0007,0x00
0x0008,0x00
0x000B,0x68
0x0016,0x02
0x0017,0xDC
0x0018,0xFF
0x0019,0xFF
0x001A,0xFF
0x002B,0x02
0x002C,0x00
0x002D,0x00
0x002E,0x00
0x002F,0x00
0x0030,0x00
0x0031,0x00
```

Procédure

3 phases :

Pre-amble

1. First, write in the preamble

   Write 0x0B24 = 0xC0

   Write 0x0B25 = 0x00

   Write 0x0540 = 0x01 (NOTE: for all new designs it is recommend that this register be written as part of the preamble. In some rare cases, omitting this write may result in a one-time LOL occurrence. However, if this issue has not occurred with your current frequency plan it is not likely to occur)
2. Wait 300 ms for Grade A/B/C/D/J/K/L/M, Wait 625ms for Grade P/E

Core registers

3. Then perform the desired register modifications

Post-amble

4. Write SOFT_RST 0x001C[0] = 1
5. Write the post-amble

   Write 0x0540 = 0x00 (NOTE: for all new designs it is recommend that this register be written as part of the post-amble. In some rare cases, omitting this write may result in a one-time LOL occurrence. However, if this issue has not occurred with your current frequency plan it is not likely to occur)

   Write 0x0B24 = 0xC3

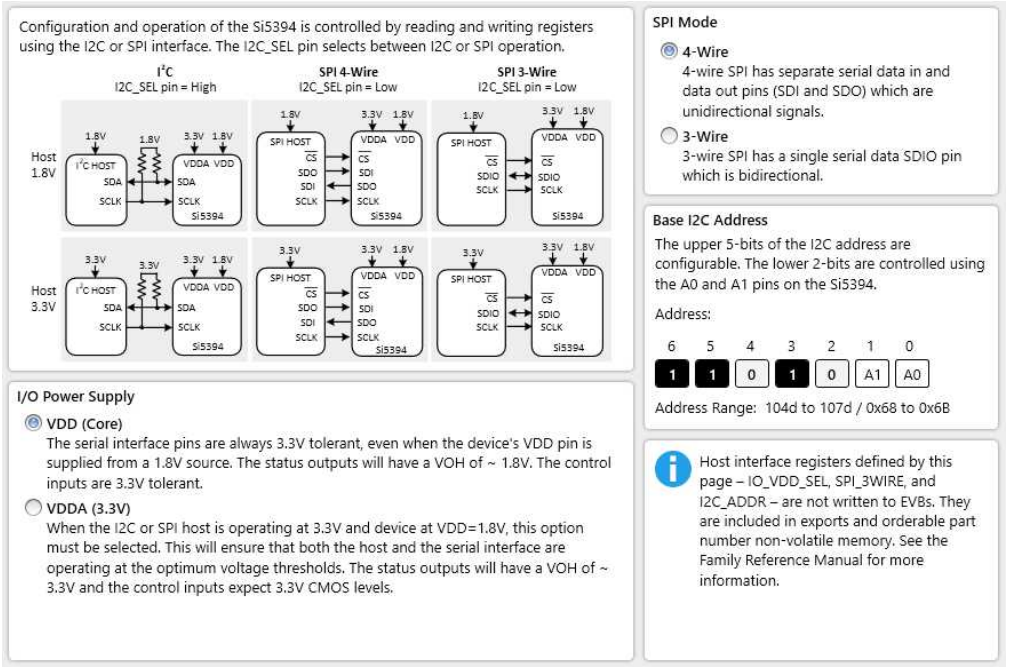
   Write 0x0B25 = 0x02

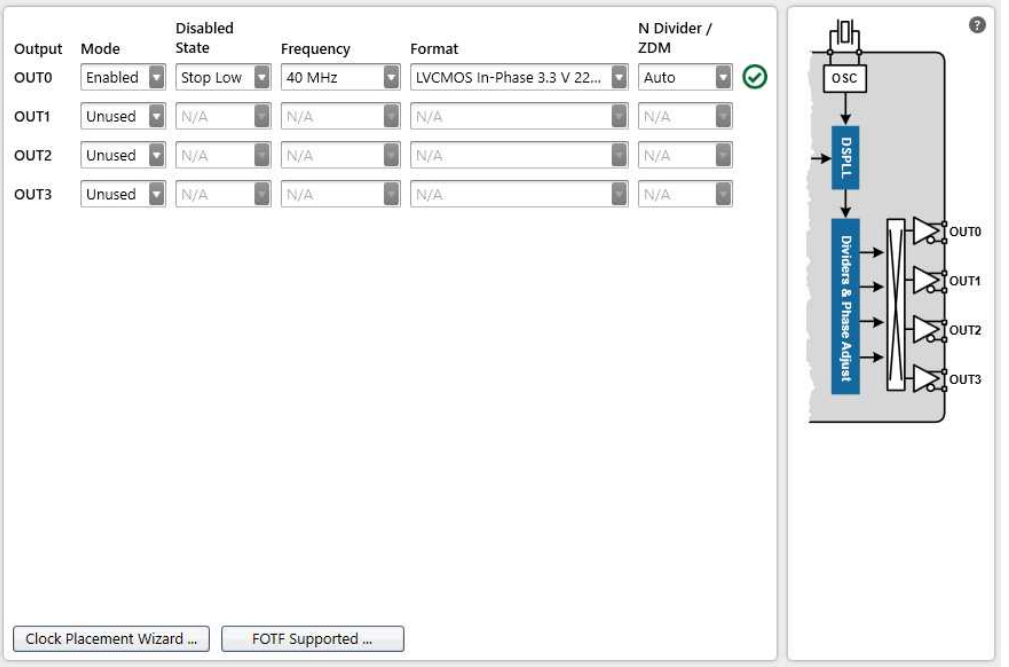
ClockBuilder Pro

11 steps

- Design Id
- I2C address
- Chip grade – Oscillator source
- Free running mode
- Output frequency
- Input source
- Jitter minimization
- Skew control
- Hitless switching
- LOS thresholds
- Interrupt sources

Pour l'execice, le fichier exporté est fourni





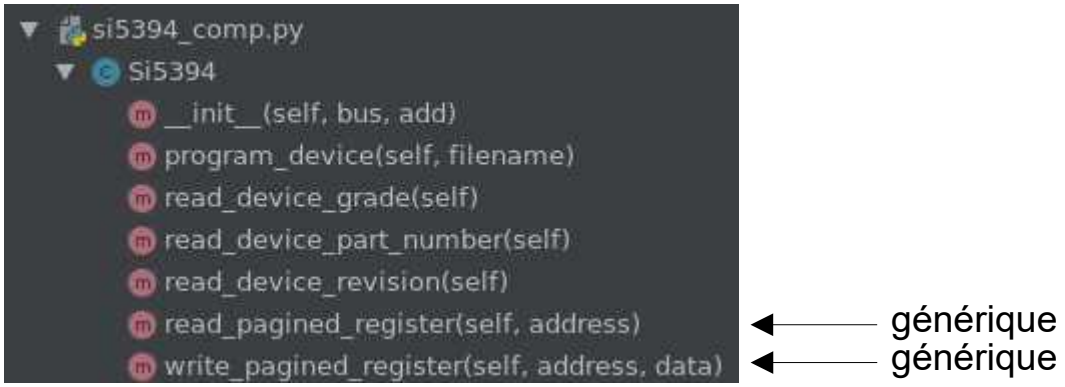
Recommendations

Ecrire des méthodes génériques d'accès aux registres

- register_read, register_write
- clear_bit, set_bit, test_bit, write_field, read_field
- etc ...

Les méthodes réalisant une fonction s'appuient sur les méthodes génériques

Exemple :



Attention au type de données de la méthode write

- Attend une liste de bytes et non un integer → utiliser data.to_bytes(1, byteorder="big")