

Hi-Rel Electronics

Mesure courant-tension

Rapport de laboratoire

Master HES-SO

Émilie Gsponer, Yann Maret

2 Mars 2016
version 1.0

Table des matières

1	Introduction.....	5
1.1	Choix du capteur.....	5
1.1.1	Principes physiques disponibles	5
1.1.2	Principe choisi.....	5
1.2	Schéma bloc de la chaîne de mesure	5
1.3	Analyse des performances attendues	5
1.3.1	Rédiger une liste de performances fonctionnelles.....	6
1.3.2	Déterminer les caractéristiques physiques du circuit à réaliser	6
2	Simulation de la mesure de courant	7
2.1	Charge variable.....	7
2.2	Décharge de la batterie	8
2.3	Courant sinusoïdal	9
3	Chaîne d'acquisition	10
3.1	Schéma électrique	10
3.1.1	Schématique de la carte capteur.....	10
3.1.2	La partie d'alimentation	11
3.1.3	La partie capteur.....	11
3.1.4	La partie connecteurs	11
3.2	Routage de la carte capteur	12
3.2.1	La mécanique du PCB	12
3.2.2	Les couches et plans	12
3.2.3	Les pistes	13
3.2.4	Les chemins de retour	13
3.3	Réalisations.....	13
3.4	Fabrication.....	13
3.4.1	L'électronique.....	13
3.4.2	Modifications.....	15
3.5	Budget de la chaîne de mesure	15
3.5.1	Liste des composants.....	15
3.5.2	Budget de masse	16
3.5.3	Budget d'espace	16
3.5.4	Budget de consommation	16
4	Conception détaillée.....	16
4.1	Dissipation de chaleur	16

4.1.1	Influence de PDiss sur TJ	18
4.1.2	Influence de TSC sur TJ	19
5	Failure Mode, effects and criticality analysis (FMECA)	20
5.1	Déterminer les modes de fonctionnement	20
5.2	Construire un arbre des défaillances	20
5.3	Déterminer les modes de défaillance et qualifier la criticité des défaillances	20
5.4	Décider des suites : amélioration de conception / limitations d'utilisation / mesures de protection	21
6	Part Stress Analysis (PSA)	21
6.1	Les hypothèses de travail	21
6.1.1	Liste des documents d'entrées : schéma et liste de pièces	21
6.1.2	Températures considérées : carte, composants	21
6.1.3	Puissances considérées : max, typ	21
6.1.4	Liste des documents applicables considérés pour les valeurs de Derating	22
6.2	La synthèse de l'analyse	22
6.3	Dimensionnement du filtre passe-bas	22
7	Worst Case Analysis (WCA)	22
7.1	Détermination des paramètres à considérer	23
7.2	Détermination des performances à analyser	23
7.3	Analyse Monte Carlo	23
7.4	Evaluation des résultats	24
8	Analyse de fiabilité (FRA)	24
8.1	Circuit de mesure à redondance froide	25
8.2	Questions	26
8.3	Nombre de pannes par milliard d'heures (Failure in Time, FIT)	26
9	Systèmes numériques	27
10	Périphérique de contrôle	27
10.1	Convertisseur A/D	27
10.1.1	Modification de l'entité	28
10.1.2	Adaptation de convertisseur pour une lecture par commande	28
10.1.3	Timing à respecter	29
10.1.4	Simulation de l'ADS1282	30
10.2	Périphérique AMBA	31
10.2.1	Modification de l'entité	31
10.2.2	Modification du contrôleur	31

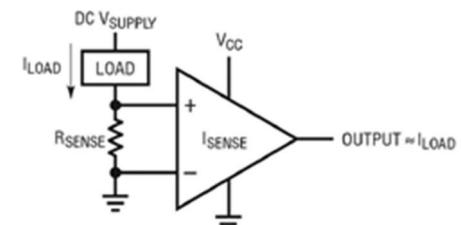
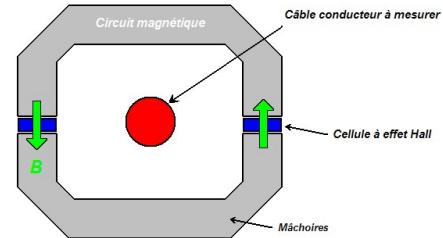
10.3	Machine d'état du contrôleur ADC.....	32
10.4	Codage de la machine d'états	32
10.4.1	Signaux de contrôle	32
10.4.2	Fiabilisation.....	33
11	Effets des radiations	33
11.1	Registre à décalage pour réception d'un signal série.....	33
11.2	Machine d'état du contrôleur ADC avec contraintes SEU.....	36
11.3	Décodeur de Hamming.....	37
11.3.1	<i>Réponse aux questions des slides</i>	38
11.3.2	Mise en commun du codage de Hamming avec correction de séquence.....	38
11.3.3	Transmission de l'erreur au registre de status	39
11.4	Simulation de l'ensemble du circuit	40
11.5	Estimation du taux de SEU par jour.....	40
12	Conception de circuit numériques	41
12.1	Alimentation FPGA	41
12.1.1	Circuit d'alimentation.....	41
12.1.2	Séquencement.....	42
12.2	Entrées / Sorties du circuit	43
13	Vérification de circuits numériques.....	43
13.1	Matrice de conformité	43
14	Tests fonctionnels.....	43
14.1	Environnement de test.....	43
14.2	Plan de test.....	44
14.3	Procédure de test	45
15	Conclusion	46

1 Introduction

1.1 Choix du capteur

1.1.1 Principes physiques disponibles

- Capteur à effet Hall :
 - Lorsqu'un courant traverse un barreau en matériau semi-conducteur (ou conducteur), et qu'un champ magnétique d'induction B est appliqué perpendiculairement au sens de passage du courant, une tension, appelée tension Hall, proportionnelle au champ magnétique et au courant, apparaît sur les faces latérales du barreau.^{1 2}
- Mesure Shunt :
 - La mesure de courant par résistance Shunt consiste à insérer une résistance de valeur connue en série avec l'alimentation et à mesurer la tension à ses bornes par mesure différentielle. La loi d'ohm permet de connaître le courant traversant la résistance.³



1.1.2 Principe choisi

Nous avons décidé de choisir la **mesure shunt**, car nous avons jugé les capteurs à effet hall trop volumineux et non pratiques car il faut que le fil d'alimentation de la batterie passe à travers le capteur pour la mesure. De plus, avec ce capteur qui mesure à l'aide d'un champ magnétique, il y a plus de chances de mesurer des perturbations qu'avec le principe choisi.

En plus du courant, notre carte devra également mesurer la tension d'alimentation.

1.2 Schéma bloc de la chaîne de mesure

Dossier de référence : */SchemaBloc*

Notre système de mesure devra se placer directement après la source d'alimentation, que ce soit par batterie ou USB. Sans cela, notre circuit ne mesurera pas tout le courant consommé. C'est pourquoi nous allons devoir refaire la carte *Slave power alimentation* afin d'y insérer notre circuit de mesure.

Afin d'avoir assez de lignes pour que le convertisseur A/D puisse communiquer avec la carte FPGA, nous avons décidé de supprimer le switch ainsi que les *user Leds* de la carte d'alimentation existante.

1.3 Analyse des performances attendues

Cette analyse liste les performances fonctionnelles (électriques, bruit, précision...) du système en construction et les caractéristiques physique des composants pour comprendre leurs interactions.

¹ https://fr.wikipedia.org/wiki/Effet_Hall

² <http://docplayer.fr/docs-images/24/4149845/images/5-0.png>

³ <http://www.electronique-mag.com/IMG/gif/3-2.gif>

1.3.1 Rédiger une liste de performances fonctionnelles

	Bruit	Résolution	Précision	Dynamique
Composant				
ADS1282	Low-Noise PGA	High Resolution: 124-dB SNR (1000 SPS)	High Accuracy: THD: -102 dB INL: 0.5 ppm	Wake up : Group delay :
MAX865	X	X	90% pour V+	Variation avec la température
MAX6350	Very Low, 1.5µVp-p Noise (0.1Hz to 10Hz)	±15mA Output Source and Sink Current	±0.02% Initial Accuracy	Ultra Low, 1ppm/°C
OP148	Low noise: 3.9 nV/ √Hz	X	Low offset voltage: 65 µV	High slew rate: 4.0 V/ µs

Remarque :

- Le bruit de la conversion AD dépend fortement du gain appliqué au convertisseur ainsi que le taux de données, voir la figure suivante.
- Recommandation pour le layout afin de diminuer le bruit du composant
 - Mount all components as close together as possible*
 - Keep traces short to minimize parasitic inductance and capacitance*
 - Use a ground plane.*
- La tension et le courant en entrée ne varient pas fortement, ce sont des tensions continues après conversion donc pas de bruit supplémentaire générer par l'amplificateur.

Table 1. Signal-to-Noise Ratio (dB)⁽¹⁾

DATA RATE (SPS)	PGA						
	1	2	4	8	16	32	64
250	130	130	129	128	125	119	114
500	127	127	126	125	122	116	111
1000	124	124	123	122	119	113	108
2000	121	121	120	119	116	111	106
4000	118	118	117	116	113	108	103

(1) $V_{IN} = 20mV_{DC}/\text{PGA}$.

La figure ci-dessus montre la qualité signal sur bruit de l'entrée du convertisseur analogique digital. Plus le gain du convertisseur est grand (64) et plus la conversion est rapide en « samples per seconde » plus la qualité du signal se détériore.

1.3.2 Déterminer les caractéristiques physiques du circuit à réaliser

Le projet complet doit être de la taille d'une cannette. Le PCB réalisé doit avoir les dimensions suivantes : 8.3x3.5x1.5mm, sans compter la hauteur des composants.

Une analyse de consommation a été faite (dans le budget du système dans l'onglet « power, components »), une puissance d'environ 242mW pour le circuit complet est à prévoir.

Pour les dimensions de la carte, les fichiers de fabrication donnent aussi des informations de hauteur. La carte passera sans problèmes dans le boîtier du CanSat avec la mécanique autour.

2 Simulation de la mesure de courant

Avec le logiciel LTSpice, nous avons pu simuler notre circuit de mesure de courant dans plusieurs cas. Selon les spécifications données, le courant consommé par l'ensemble du circuit est aux alentours de 130 mA. Nous avons volontairement gardé une marge, car on ne connaît pas encore la consommation de l'amplificateur et du convertisseur A/D. La formule suivante a été utilisée pour le dimensionnement :

$$V_{current} = I_{load} * R_{shunt} * \frac{R_1}{R_3}$$

Nous avons dimensionné notre amplificateur avec un gain de 20 et choisi une résistance shunt de 0.1 ohm. Cela nous donne une tension de 4V pour un courant de charge de 200 mA. Comme la tension d'alimentation est de 5V et que l'ampli est rail-to-rail, on peut théoriquement mesurer jusqu'à 250 mA.

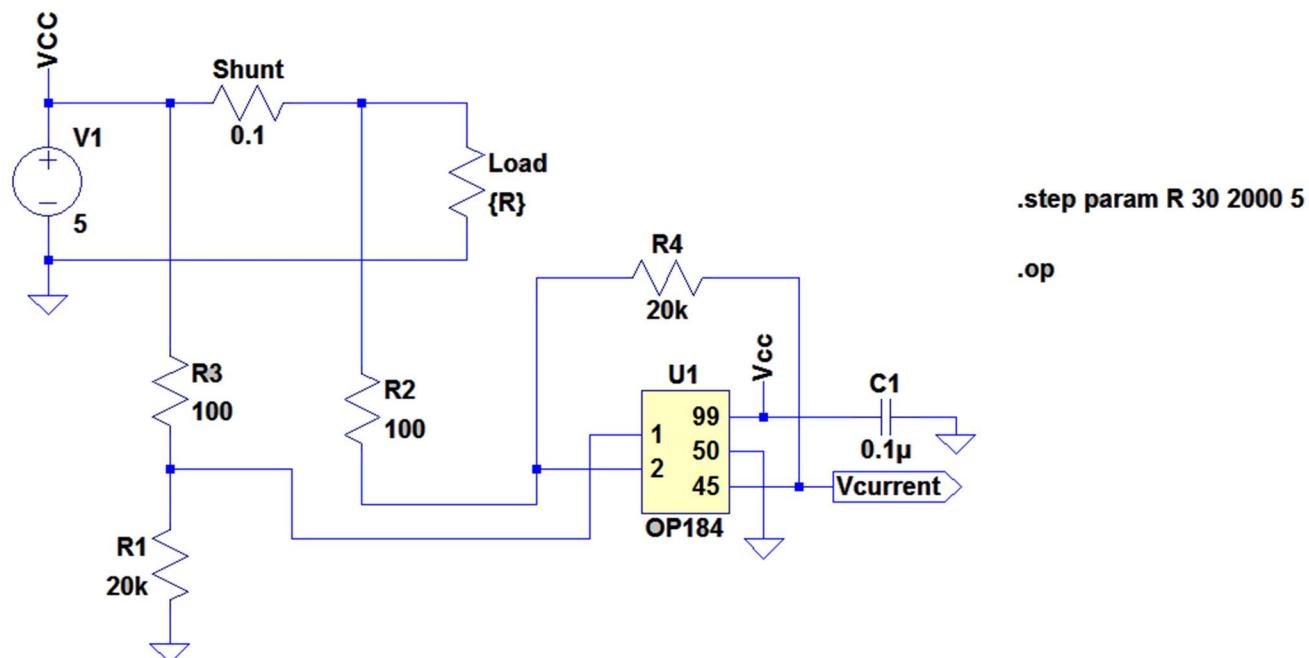


⁴ Le composant de notre Amplificateur OP184 a été importé dans LTSpice. Il se peut qu'il faille le réimporter pour pouvoir reproduire nos simulations. C'est pourquoi nous avons fourni ci-dessous des captures d'écran pour les différents tests.

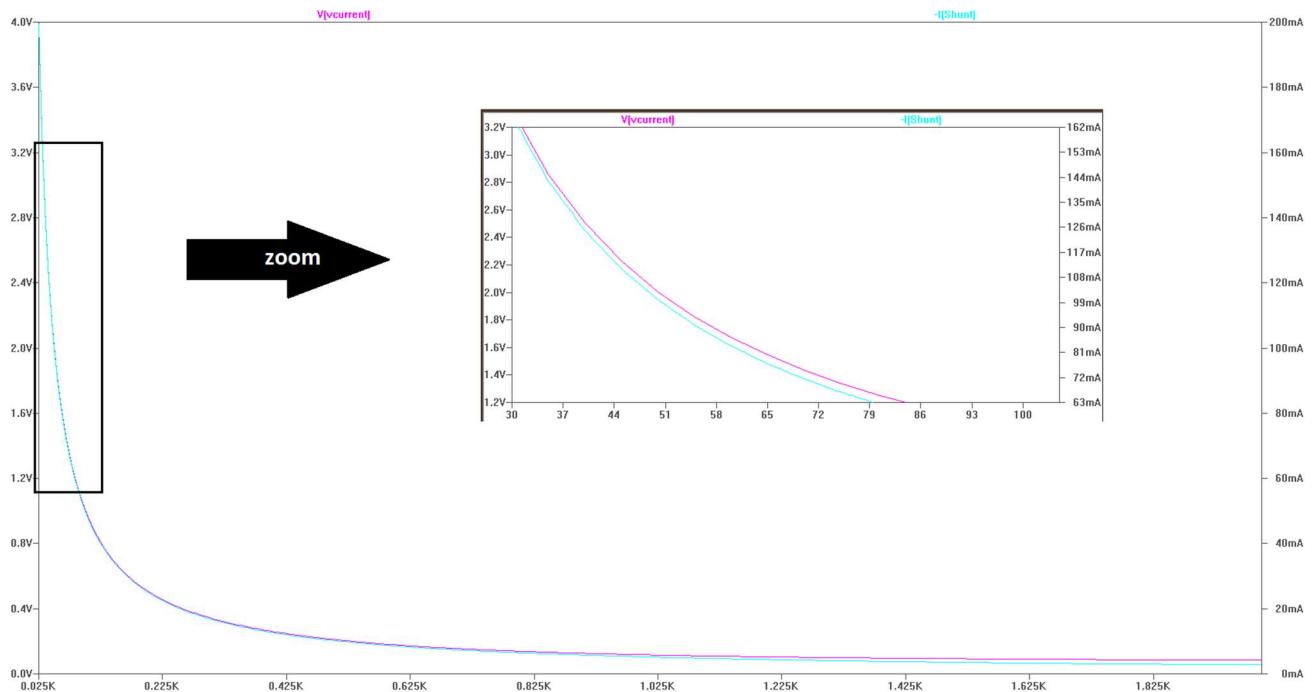
2.1 Charge variable

Dossier de référence : */Simulation/chargeVariable*

Dans cette simulation, la charge du circuit varie de façon à avoir un courant consommé variable. Le but est d'observer si la tension en sortie de l'amplificateur suit correctement la courbe de courant.



⁴ Source de l'image : http://images.all-free-download.com/images/graphicthumb/sign_alert_98028.jpg

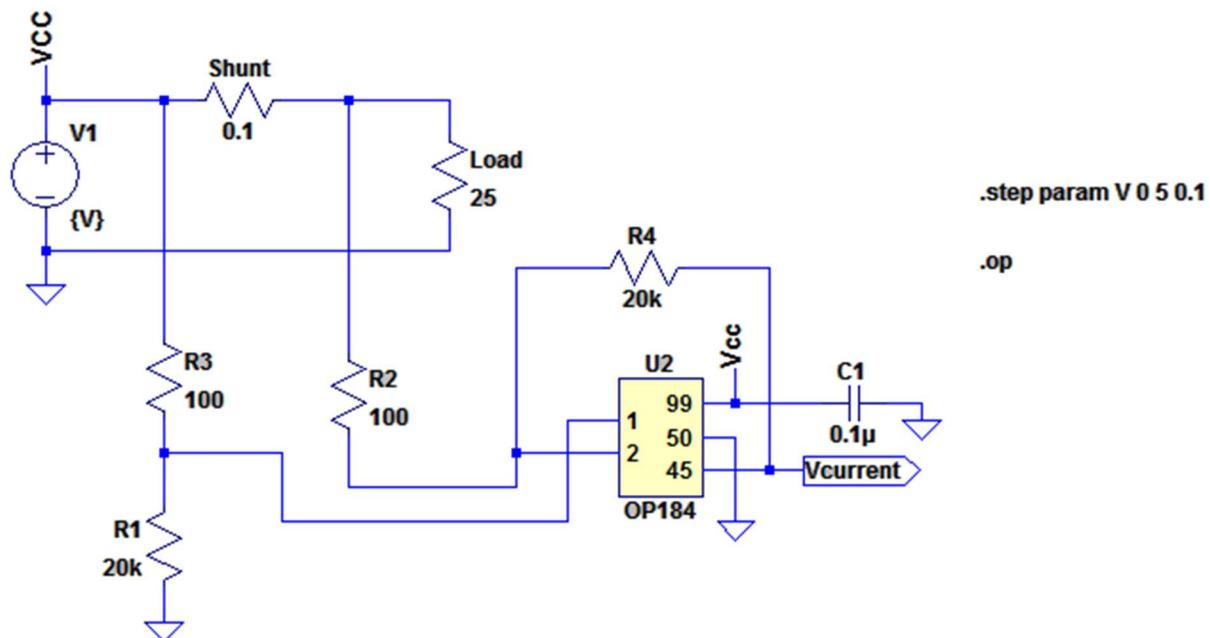


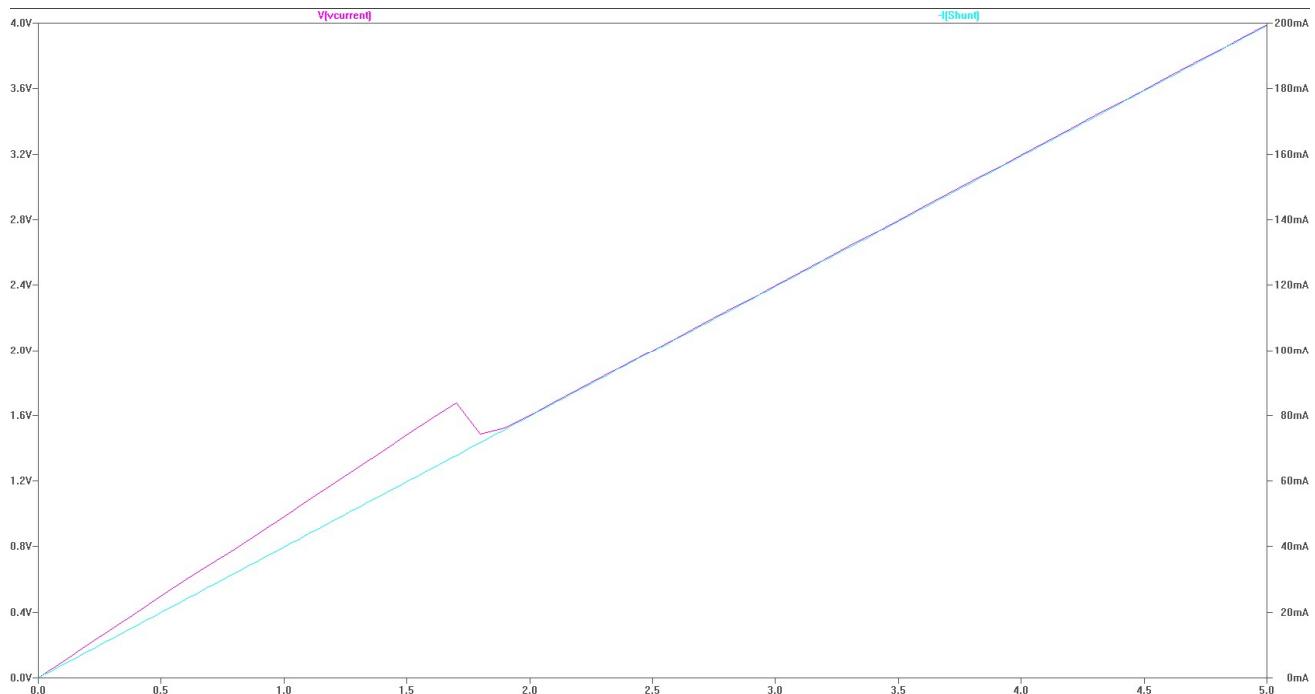
On constate que notre ampli op transforme correctement le courant du shunt en tension jusqu'aux alentours de 5 mA. En dessous de cette valeur, la tension fournie reste constante à 100 mV.

2.2 Décharge de la batterie

Dossier de référence : [/Simulation/dechargeBatterie](#)

Dans cette simulation, on observe le comportement du système lorsque la tension d'alimentation diminue.



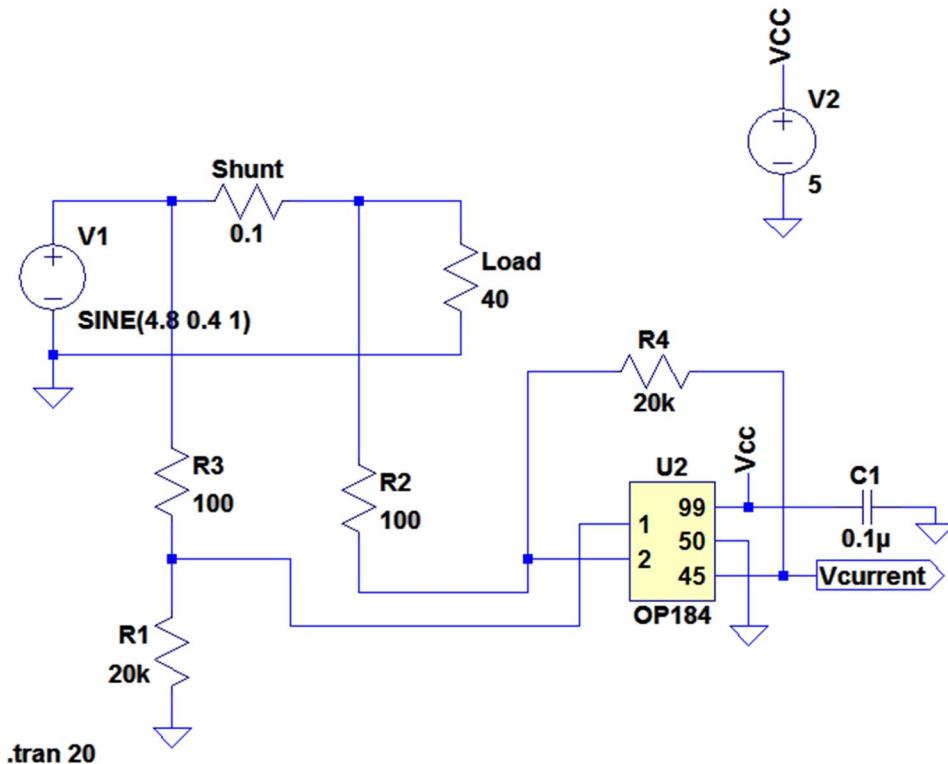


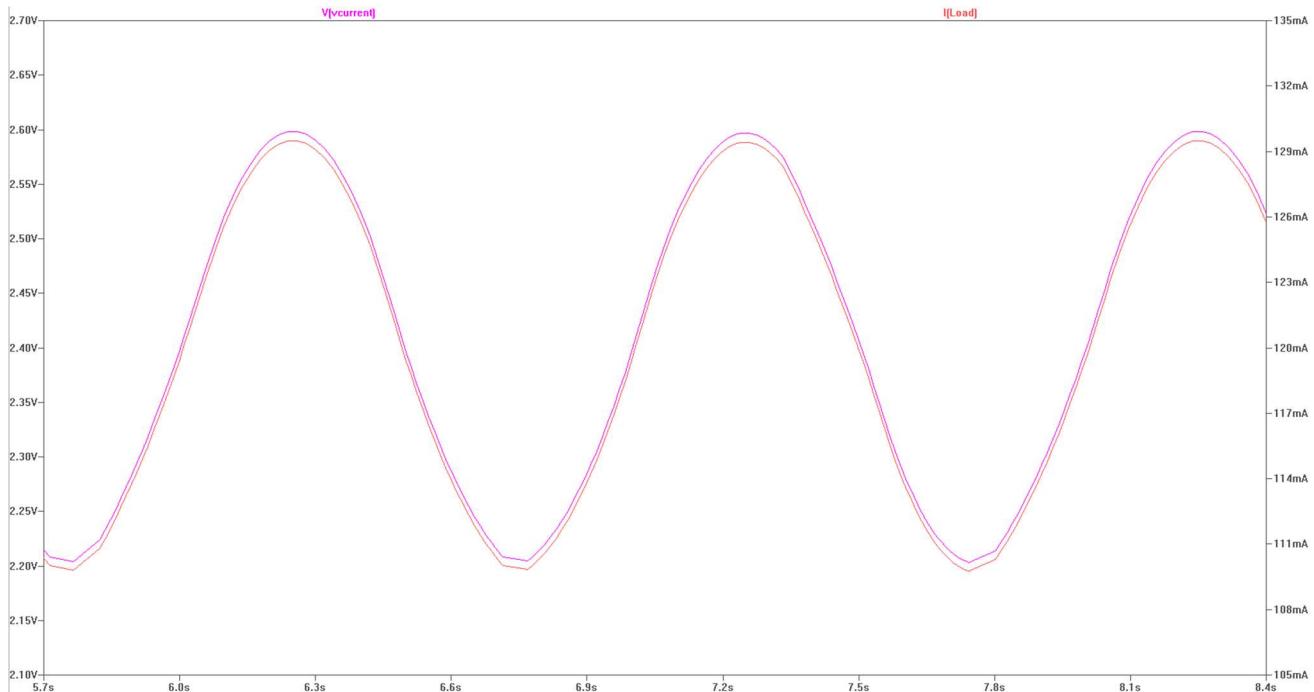
On constate qu'il se passe un phénomène étrange en dessous de 2V, mais que le circuit mesure correctement au-dessus de ce seuil de tension d'alimentation.

2.3 Courant sinusoïdal

Dossier de référence : [/Simulation/courantSinus](#)

Dans la mesure de courant qui nous a été fournie, on a pu observer que le circuit réel n'avait pas une consommation de courant stable, mais qu'elle oscillait entre 110 et 130 mA. Nous avons reproduit cette situation.





Cette simulation nous a permis de constater qu'il nous fallait absolument une référence stable de tension 5V pour alimenter notre amplificateur, sinon la mesure n'est pas correcte. On voit ci-dessus que la mesure suit correctement l'oscillation du courant consommé.

- ✓⁵ Ces différentes simulations ont permis de valider le principe de mesure de courant et le choix de l'amplificateur opérationnel.
- ✗⁶ Nous n'avons pas trouvé de modèle LTSpice de notre convertisseur A/D. Nous n'avons donc pas pu simuler entièrement le comportement de notre circuit de mesure.

3 Chaîne d'acquisition

3.1 Schéma électrique

Dossier de référence : */Circuitmaker*

Cette section explique l'organisation de la schématique de la carte capteur ainsi que le design appliqué au routage du PCB. Tout a été fait avec CircuitMaker.

Il est possible de trouver la schématique, le PCB, les composants utilisés pour le circuit ainsi qu'une vue 3D de l'électronique sur le cloud de CircuitMaker en recherchant le projet nommé « HIREL » dans l'onglet « Projects » puis « Open Project ». Il est ensuite normalement possible de visualiser le projet.

3.1.1 Schématique de la carte capteur

La schématique de la carte est composée d'une partie d'alimentation, une partie entrée-sortie et une dernière partie pour la connexion avec les autres cartes.

⁵ Lien : http://www.experimentmultimedia.ch/ressources/graphisme-symboles-logos/symboles-1/symbole-vu/image_large

⁶ Lien : <https://esgmsc.files.wordpress.com/2014/04/croix-rouge.png>

Vu que nous devons refaire la carte d'alimentation pour que notre mesure fonctionne, nous devons fournir diverses références de tension : 5V et 3.3V pour alimenter les autres cartes et les lignes de communication logique avec le convertisseur AD.

Les différentes parties de la schématique sont décrites ci-dessous :

3.1.2 La partie d'alimentation

- Un convertisseur DC/DC pour la création d'un 3.3V.
- Un composant FTDI pour l'alimentation de la carte et la programmation de la FPGA.

Remarques : Cette partie est reprise de la carte d'alimentation actuelle du CanSat.

3.1.3 La partie capteur

- Une alimentation stabilisée pour l'alimentation de la partie analogique du convertisseur AD.
- L'amplificateur pour faire la mesure du courant.

3.1.3.1 Référence de tension

Cette référence de tension est faite grâce à un doubleur de tension à partir du 5V d'entrée (batterie/USB).

Une référence de tension est nécessaire pour les entrées REF du convertisseur AD afin d'avoir des conversions non erronées. La référence de tension garanti une tension de 5V +/- 1mV.

Pour un convertisseur 24bits : $2^{24} = 16\ 777\ 216$

1mV = 3355 LSB \rightarrow 12bits de bruit, donc sur 24bits, il ne reste plus que 12bits de signal utile.

Pour notre application de housekeeping (mesure de courant-tension) la moitié de la résolution nous suffit, les 12 MSB sont largement suffisants. Le courant en fonctionnement n'est consommé que par la FGPA et le niveau de la tension de la batterie ne devrait pas trop osciller.

3.1.3.1.1 L'amplificateur (mesure de courant)

Pour la mesure du courant avec l'amplificateur opérationnel, le montage est repris des simulations effectuées précédemment. Une résistance shunt est placée en série à l'alimentation. Le reste du circuit est connecté après cette résistance afin de mesurer le courant de l'ensemble du circuit.

La mesure de tension est faite depuis le header qui aiguille la source d'alimentation soit depuis la batterie, soit depuis l'USB.

3.1.4 La partie connecteurs

- Un connecteur pour la batterie
- Un connecteur micro USB pour l'alimentation et la programmation de la FPGA
- Un header trois broches pour la sélection de la source d'alimentation (batterie/USB)
- Deux connecteurs pour l'assemblage et le contact avec le couvercle et le fond (TOP-BOTTOM).

Les connecteurs (TOP-BOTTOM) ont le câblage suivant :

3.1.4.1 Le connecteur TOP

Le connecteur « Top » connecte le couvercle avec la carte capteur. Les principaux signaux sont les lignes de communication SPI entre le convertisseur AD et la FPGA.

2	SCLK	SPI
4	DIN	
6	DOUT	

8	DRDY	
10	GND	
12	VCC5	

3.1.4.2 Le connecteur BOTTOM

Le connecteur « Bottom » sert de liaison avec la carte FPGA, les lignes TX-RX sont pour communication. Quant aux autres signaux, ils vont sur le convertisseur AD.

1	TXD	UART
3	RXD	
5	RESET	Sampling
7	CLK	
9	GND	
11	VCC33	

Remarques : Pour l'alimentation de l'amplificateur opérationnel ainsi que la partie analogique du convertisseur analogique digitale, il serait préférable de mettre en place un régulateur de tension dédié et de réserver l'utilisation de la référence de tension pour les entrées REF et CAP du convertisseur.

La référence de tension fournit un courant de 15mA en sortie, elle ne supporte pas bien les variations d'impédance.

L'élévateur de tension monte la tension à 10V, la tension de fonctionnement du circuit est de 5V. Une petite puissance est perdue sur la référence de tension vu que le courant est faible dans l'amplificateur ainsi que dans le convertisseur AD.

3.2 Routage de la carte capteur

3.2.1 La mécanique du PCB

Les dimensions du PCB sont de 83x35mm.

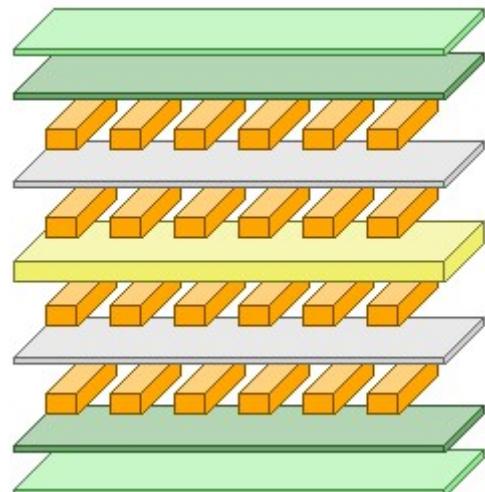
Les composants sont placés sur le « Top » de la carte. Cinq condensateurs et une résistance sont montés sur le « Bottom ». Les connecteurs TOP et BOTTOM, sont placé sur la tranche du PCB. Un header peut être monté avec un pitch de 30 mils.

3.2.2 Les couches et plans

Le routage est confectionné sur un PCB 4 couches.

Comme le montre la figure sur la droite, le PCB est organisé de la façon suivante :

Nom couche	Type	Matériaux	Epaisseur [mil]
Top Overlay	Overlay		
Top Solder	Solder Mask/Coverlay	SurfaceMaterial	0.4
Top Layer	Signal	Copper	1.4
Dielectric 1	Dielectric	Prepreg	14
Signal Layer 1	Signal	Copper	1.417
Dielectric 3	Dielectric	Core	28
Signal Layer 2	Signal	Copper	1.417
Dielectric 2	Dielectric	Prepreg	14
Bottom Layer	Signal	Copper	1.4



Bottom Solder	Solder Mask/Coverlay	SurfaceMaterial	0.4
Bottom Overlay	Overlay		

L'épaisseur totale du PCB est donnée comme étant 1.59mm. Quatre plans sont répartis sur les différentes couches :

couche	plan
1	GND
2	5V
3	3.3V
4	GND

Sur les couches de « Top » et « Bottom » en plus du plan GND, des signaux connectent des composants entre eux, de même que sur la couche du 3.3V. Seul le 5V possède son propre plan sans interruption par des pistes.

3.2.3 Les pistes

L'épaisseur des pistes est pour la grande majorité des signaux 20 mils quelques-uns sont en 10 mils. Pour les signaux d'alimentation, des pistes de 30 mils sont utilisés, quelques plans isolés sont également employés.

3.2.4 Les chemins de retour

Après vérification, les chemins de retour pour les signaux peuvent emprunter des chemins relativement proches des lignes d'où ils viennent. Le plan clearance est fixé à 20 mils et la hauteur max entre les deux couches Top et Bottom est de 31mils. Les signaux peuvent emprunter les plans de masse du Top et du Bottom. Le plan de masse du Bottom est nettement moins saturé en signaux.

3.3 Réalisations

Le PCB doit être réalisé par une entreprise spécialisée. Il nécessite l'assemblage de 4 couches, ainsi que la mise en place de trous métallisés. Certaines contraintes de distance peuvent ne pas être garanties.

3.4 Fabrication

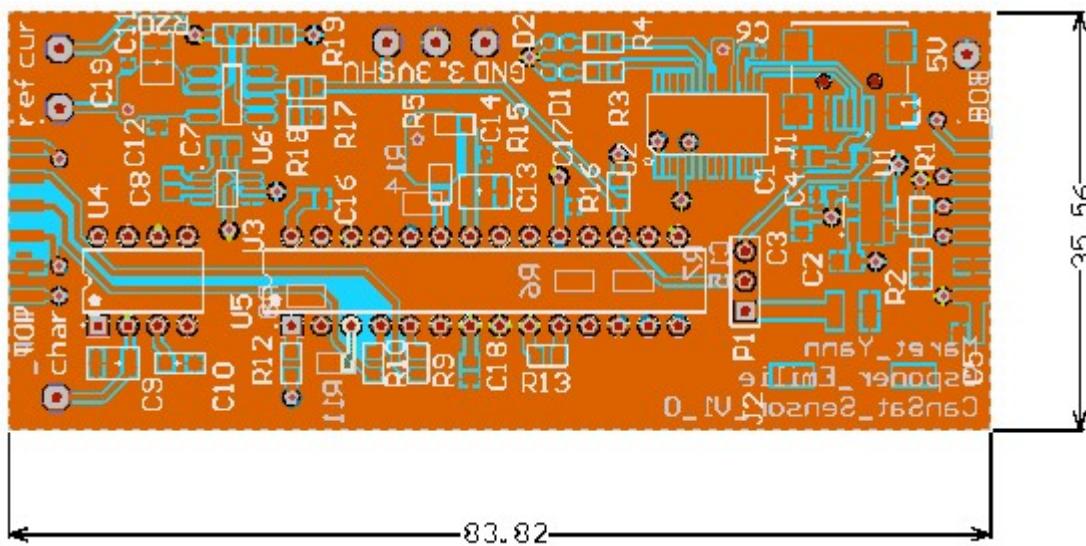
Un devis d'« EuroCircuit » peut être fait sur leur page web, pour deux exemplaires de cette carte, la fabrication reviendrait à 122Euro au total.

Quelques informations sont à fournir, concernant les couches, les épaisseurs, les dimensions du PCB. Pour éviter des prix exorbitant, la matière du PCB reste du FR-4, les vias sont traversant sur toutes les couches et les composants sont seulement montés sur les deux couches visibles de PCB (pas de composants internes).

Material		Buildup	
Board thickness	1.55 mm	Material Tg	145-150 °C
Outer layer copper foil	18 µm (end +/-35 µm)	Inner layer copper foil	35 µm
Reversed buildup	<input type="checkbox"/>	Special buildup	<input type="checkbox"/>
Inner layer core thickness	Standard	Blind/Buried via runs	0
Extra press cycles	0		

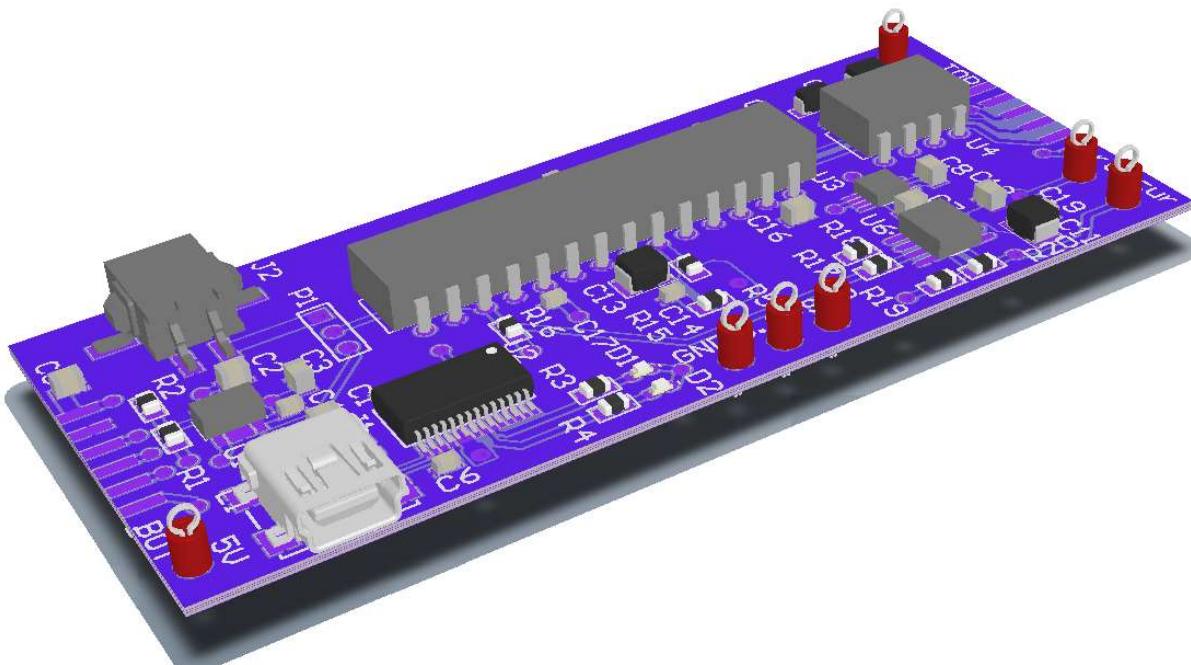
3.4.1 L'électronique

Voici le PCB de la carte électronique vu du Top :



Deux images montrent les couches « Top » et « Bottom » de la carte capteur.

Couche top :

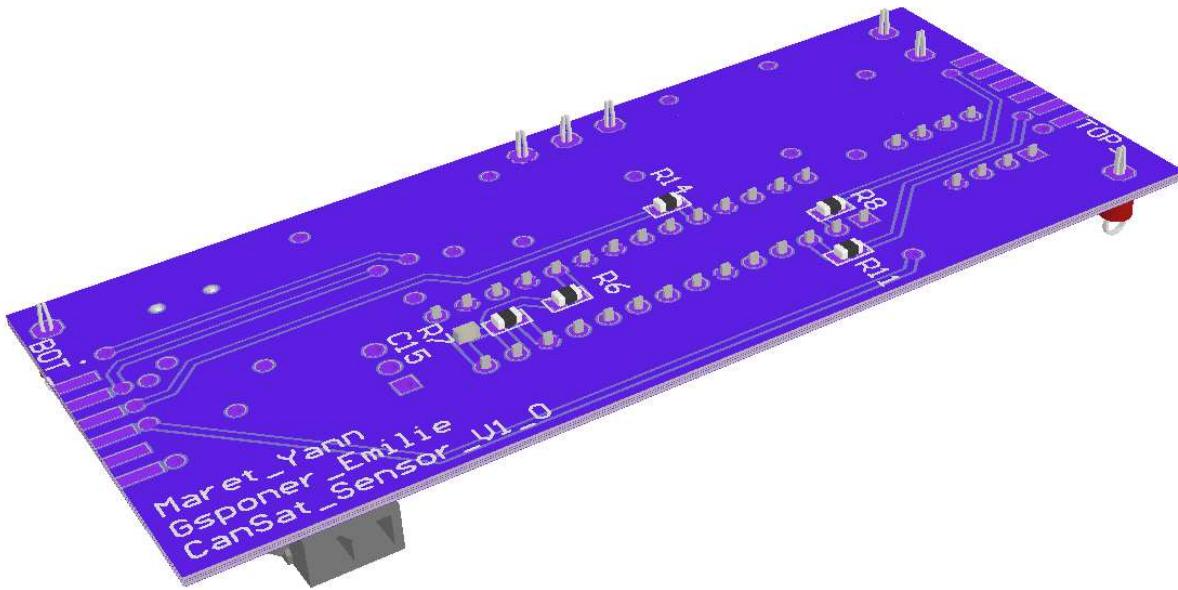


Sur cette couche on y retrouve les footprint des connecteurs « Top » et « Bottom », les autres connecteurs, les composants pour les mesures et conversion des données. Des pointes de test sont montées à titre de debug uniquement :

position	mesure
Bas- gauche	5V in
Milieu gauche	GND
Milieu milieu	3.3V
Milieu droite	SHUNT
Droite bas	Mesure courant
Droite milieu	5V_REF

12 Avril 2016

Droite haut	Doubleur de tension 10V
-------------	-------------------------

Couche bottom :

Sur la couche de bottom, une annotation est inscrite pour signer la carte et afficher le numéro de version.

3.4.2 Modifications

Il faudrait mettre un convertisseur DC/DC supplémentaire afin de fournir les tensions d'alimentation du convertisseur A/D ainsi que de l'amplificateur. Il serait possible de placer cette électronique sur la couche « Bottom ».

Une autre alternative pour gagner de la place sur le PCB est de changer le type de boîtier de l'ADC pour passer à un boîtier SMD. Une alternative de design est de mettre les vias dans les pads des composants avec recouvrement.

Par manque de temps, seul le design a été fait sans fabrication industrielle. Après des tests et simulations, puis une relecture de la schématique et du routage par M. Joseph Moerschell, la carte électronique est fonctionnelle et pourrait être commandée, soudée, programmée et testée sur le CanSat.

Dans une deuxième version, il faudrait faire une alimentation séparée des composants et ne plus utiliser le 5V_REF.

3.5 Budget de la chaîne de mesure

Dossier de référence : /BudgetChaineMesure

3.5.1 Liste des composants

La liste des composants du circuit a été incluse dans le fichier Excel sous l'onglet « Component ».

Nous avons essayé au maximum de choisir des composants respectant la plage de température en vigueur pour le spatial, soit -55°C à 125°C. Une grande partie des composants ne sont pas garantis pour cette plage de température, ils vont plutôt de -40°C à 85°C. Cela n'est pas problématique pour le CanSat qui va normalement passer très peu de temps dans l'espace.

 Le FTDI avec ses deux LEDs et ses autres petits composants pourront être enlevés avant d'envoyer le CanSat dans l'espace. Ces composants sont uniquement utiles pour la programmation de la FPGA.

Avec tous nos composants, nous arrivons à un prix total de **265 CHF**. La plus grande partie du prix étant dédiée au convertisseur analogique-digital.

3.5.2 Budget de masse

Nous n'avions pas fixé de contrainte concernant le poids de la carte. Nos composants ont un poids total de **12g**, ce qui est raisonnable. Avec l'estimation du poids du PCB et de la soudure, on passe à un poids estimé à **28 g**. Une autre estimation a été faite en multipliant l'espace occupé par le composant avec l'épaisseur du PCB et sa masse volumique, on obtient un poids de **22 g**.

3.5.3 Budget d'espace

La taille du PCB nous est imposée, nous avons le droit à une carte double faces de 35mm sur 83mm. Cela donne une surface (Top et Bottom) de **5'810 mm²**.

En additionnant l'espace pris par tous les composants, on atteint une surface de **3'146 mm²**. Cette valeur est approximative, car un ratio a été ajouté à chaque composant.

-  On peut raisonnablement s'attendre à ce que tous nos composants puissent être placés sur la carte et routés.

3.5.4 Budget de consommation

La consommation du circuit avec la FPGA en fonctionnement nous a été donnée et est de 130mA. L'estimation de consommation de la carte d'alimentation a été estimée dans le pire des cas à **65mA** avec une puissance totale de **242mW**.

Nous avons également estimé la consommation de la carte d'alimentation actuelle à 50mA, notre circuit de mesure et de conversion analogique digital consomme donc 15mA. Avec ces informations on peut donc estimer la consommation de la carte FPGA à $130\text{mA} - 50\text{mA} = 70\text{mA}$.

Si l'on met tous ces éléments ensemble, nous avons donc une consommation totale de l'alimentation et de la FGA de $65\text{mA} + 70\text{mA} = 135\text{mA}$. Cela fait que le shunt dissipe une puissance de $0.1\text{ohm} \cdot 135\text{mA}^2 = 1.8\text{mW}$

-  La résistance shunt que nous avons choisie pour la mesure du courant de toutes les cartes peut supporter une puissance maximale de 500 mW. Avec les calculs ci-dessus, on peut valider que la résistance est bien choisie.

4 Conception détaillée

4.1 Dissipation de chaleur

Dossier de référence : */DissipationChaleur*

Nous avons décidé faire notre analyse de chaleur sur le chip FTDI. Avec ses 75mW, c'est le composant qui consomme le plus. On analysera comment la chaleur va se dissiper au travers du PCB. Une simulation est disponible dans le dossier de référence.

Le datasheet du FTDI nous fournit la résistance thermique de contact JC (jonction-case). Il donne également la résistance JA (jonction-air), mais elle n'est pas utile dans ce contexte, car la dissipation par convection et radiation est négligeable dans l'espace. On a bien plus de dissipation par conduction.

Item	Material	Dimensions	Mass	Thermal conductivity [W/(m·K)]	Thermal capacity [J/(K·kg)]
FT232RL	n/a	7.8mm x 10.2mm x 2mm	2268mg	component specific	n/a
PCB conducting layers	Cu, $\rho = 8.9\text{mg/mm}^3$	35mm x 83mm, 4 layers 35μm	$m_{\text{Cu},35} = 90.5\text{mg}$ (35μm layer, fill factor 0.1) $m_{\text{Cu,plane}} = 301\text{mg}$ (35μm layer, fill factor 0.85)	$k_{\text{cu}} = 190$	$C_{\text{cu}} = 380$
PCB substrate	FR4, $\rho = 2.5\text{mg/mm}^3$, prepreg material, $\rho = 1.5\text{mg/mm}^3$	35mm x 83mm 1 FR4 of 150 μm+ 2 prepreg of 135 μm	$m_{\text{PCB}} = 1089 \text{ mg FR4} + 1177 \text{ mg Prepreg} + 90.5 \text{ mg Cu} = 2.4\text{g}$	$k_{\text{FR4}} = 0.85$	$C_{\text{FR4}} = 830$

Avec ce tableau, nous pouvons calculer les résistances thermiques et capacitives.

$$\text{Le calcul est le suivant : } R_{th} = \frac{\text{Épaisseur}}{\text{Surface} * \text{Thermal conductivity}}^7$$

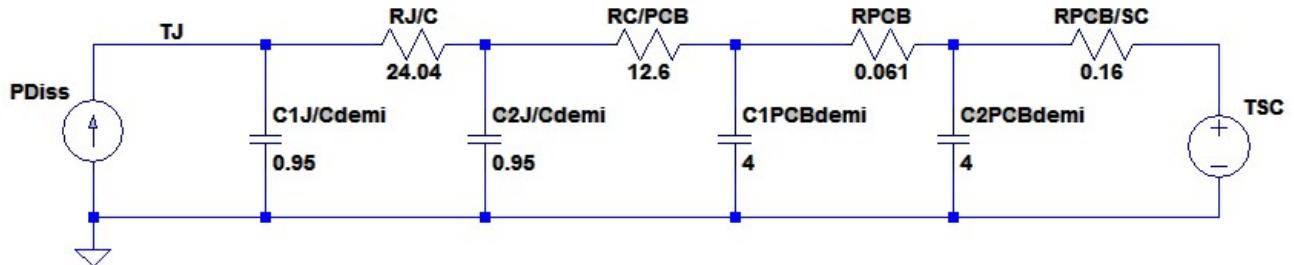
Item	Épaisseur	Surface	Résistance thermique	[K/W]
FT232RL JC	n/a	n/a	$R_{J/c}$	24.04
Contact case-PCB Estimated $R_c \approx 1\text{mK}\cdot\text{m}^2/\text{W}$	n/a	7.8mm x 10.2mm	$R_{C/PCB}$	12.6
PCB	Cu, 4*35 μm FR4, 1*150 μm	Cu, 35mmx83mm FR4, 35mmx83mm	R_{PCB}	0.061
Contact PCB-Spacecraft Estimated $R_c \approx 1\text{mK}\cdot\text{m}^2/\text{W}$	n/a	35mm x 83mm x2+ (70mm+166mm)*2mm	$R_{PCB/SC}$	0.16

$$\text{Le calcul est le suivant : } C_{th} = \text{Masse} * \text{Thermal capacity}$$

Item	Masse	Condensateur thermique	[J/K]
FT232RL	2.268 g (Estimation 0.85 J/(K·g))	$C_{J/A} \text{ et } C_{J/C}$	1.9
PCB	2.4 g PCB+11 g components (Estimation for complete PCB with components: $C_{PCB,\text{total}} = 0.6 \text{ J/(K·g)}$)	C_{PCB}	8

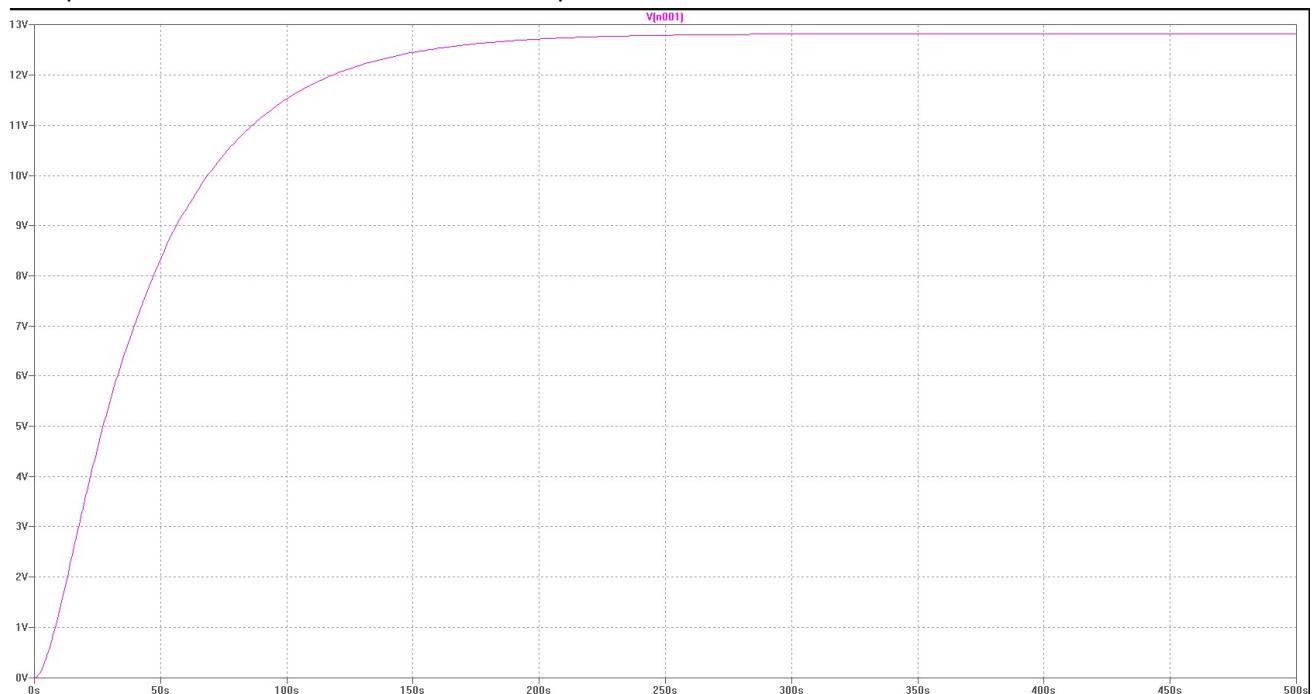
⁷ Source Wikipedia : https://fr.wikipedia.org/wiki/Conductivité_thermique

Nous obtenons le schéma équivalent suivant :

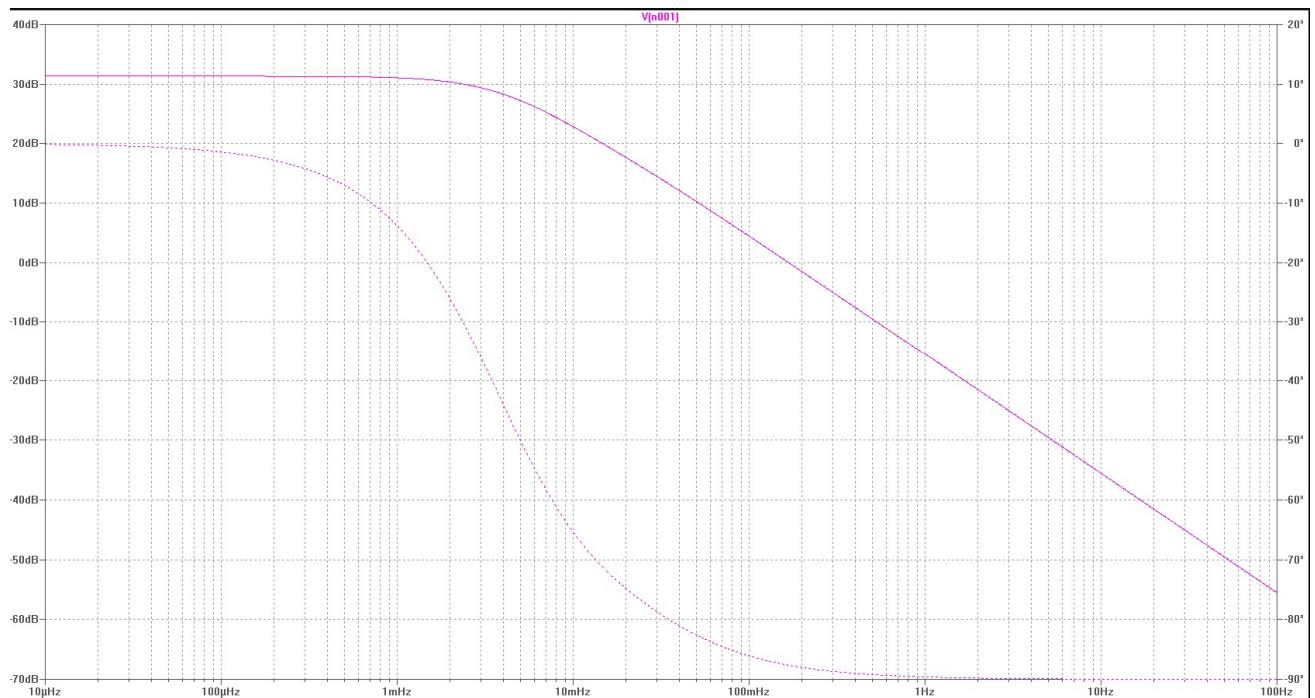


4.1.1 Influence de PDiss sur TJ

Comportement de TJ lors d'un saut de sa dissipation PDiss de 1W :

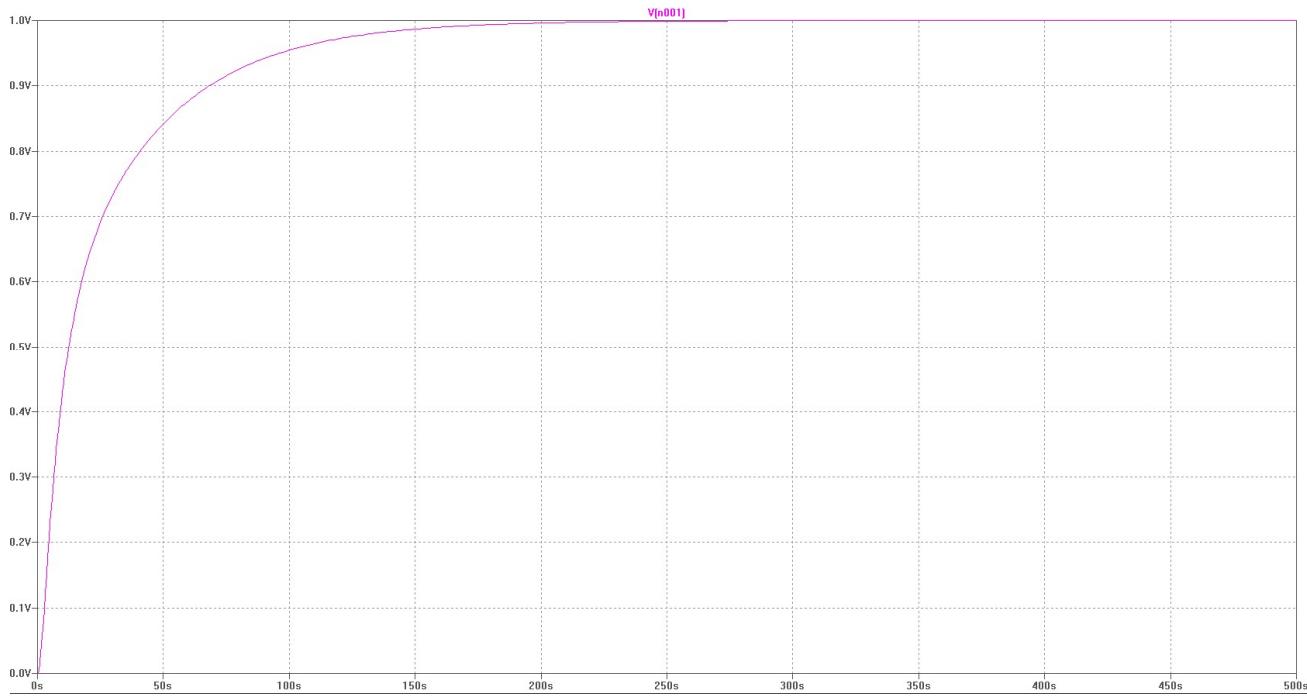


Réponse en fréquence de TJ par rapport à sa dissipation PDiss :

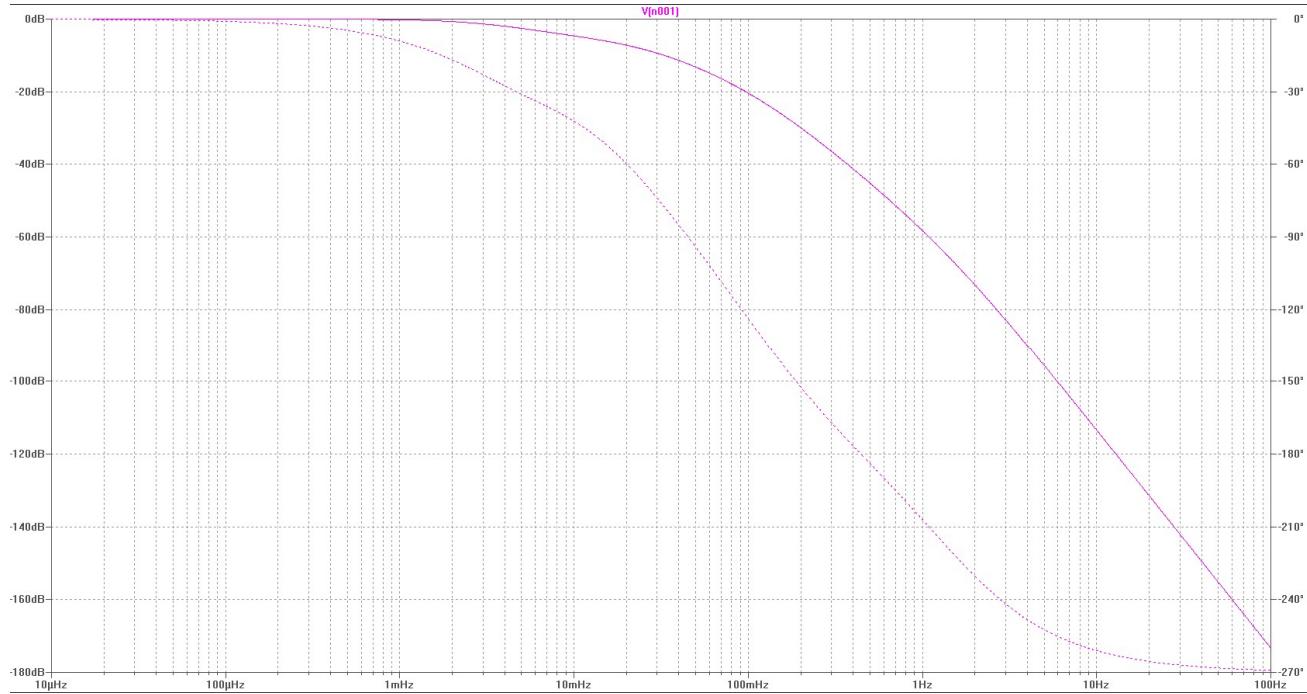


4.1.2 Influence de TSC sur TJ

Comportement de TJ lors d'un saut de température TSC de 1K



Réponse en fréquence de TJ par rapport à la température TSC :



La réponse en fréquence et le comportement par rapport à un saut donnent les mêmes informations. On peut voir qu'il faut environ 300 secondes, soit 5 minutes pour éléver la température de la jonction de 1 Kelvin. Suite à un saut de la puissance de 1 W, la température s'élève de 13 Kelvins en 5 minutes.

On remarque que la température du composant varie plus rapidement que l'exemple des slides, cela est dû au fait que le circuit présenté est beaucoup plus grand que le nôtre et est dans un boîtier.

5 Failure Mode, effects and criticality analysis (FMECA)

Dossier de référence : /FMECA

Cette analyse détermine les modes de défaillance, leurs effets et le risque associé. La FMECA s'appuie sur un tableau, un arbre des défauts et des échelles de criticité et de sévérité.

Cette analyse est faite selon la norme MIL-STD-1629A.

5.1 Déterminer les modes de fonctionnement

Les modes de fonctionnement normal du CanSat sont les suivants :

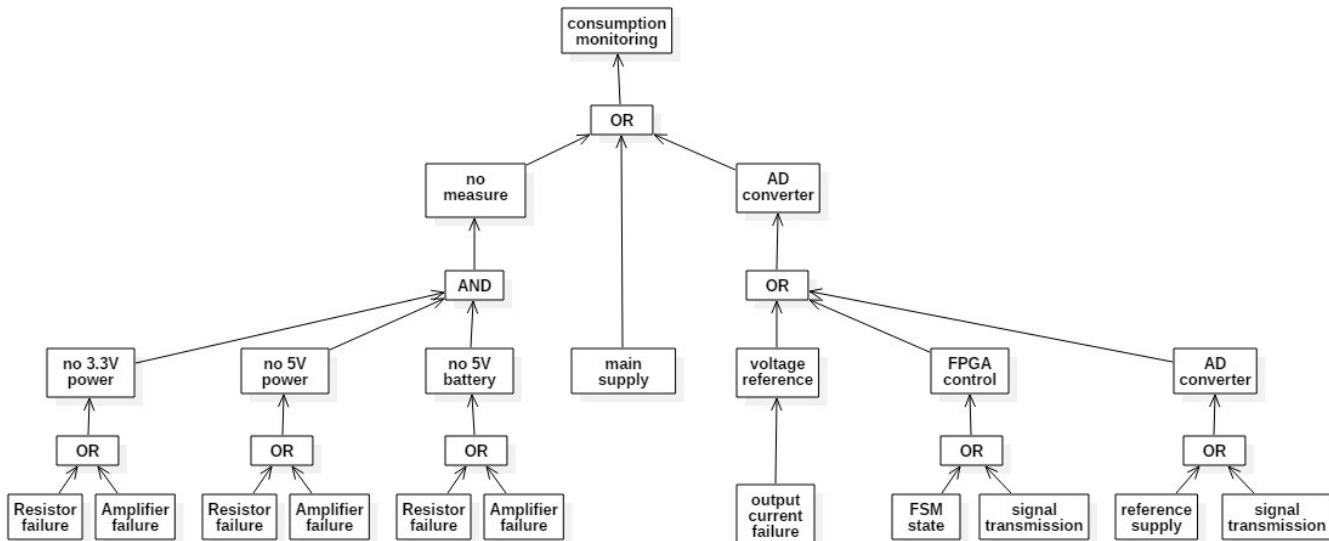
Le mode de fonctionnement global est que toutes les sous-parties du CanSat sont opérationnelles. Avec les alimentations, les capteurs, le convertisseur, le protocole de communication et la logique numérique (FPGA) fonctionnelles.

Les modes de fonctionnement sont listés dans le tableau suivant :

Blocs fonctionnels	Mode de fonctionnement
Alimentations	Tension 3.3V et 5V en sortie
Capteur courant-tension	Mesure du courant au travers de l'ampli op
Échantillonnage courant-tension	Le convertisseur échantillonne et fournit les valeurs
Communication partie logique	Communication entre la carte capteur et celle de la FPGA
FPGA	Bon traitement de donnée reçu sur les ports d'entrées

5.2 Construire un arbre des défaillances

La figure suivante montre l'arbre de défaillance du système de mesure courant-tension avec shunt :



5.3 Déterminer les modes de défaillance et qualifier la criticité des défaillances

Le tableau dans le fichier « fmeca_table » liste les modes de défaillances ainsi que leur criticité.

En résumé, la mesure de courant-tension à la source de l'alimentation de la carte est très délicate, comme le montre le résultat. Si soit la shunt, soit l'amplificateur venaient à être défectueux, cela serait dévastateur pour le reste de la carte. La tension d'alimentation du reste des composants ne pourrait plus être garantie et le système serait alors totalement hors de contrôle.

5.4 Décider des suites : amélioration de conception / limitations d'utilisation / mesures de protection

Une bonne solution serait de faire de la redondance au niveau de l'alimentation de la carte en multipliant les sources de tensions. Il est bien sûr important de ne pas surcharger le CanSat avec une autre électronique trop gourmande en énergie sous peine de dissiper un courant trop important à l'intérieur du shunt.

Comme autre mesure de protection, il serait envisageable de mettre une diode en inverse à la sortie du bloc d'alimentation. Une autre alimentation pourrait à son tour prendre la relève et fournir la nouvelle tension d'alimentation de la carte.

6 Part Stress Analysis (PSA)

Dossier de référence : /PSA et /datasheet

Cette analyse permet de sélectionner des composants avec une certaine marge de stress afin de garantir une certaine fiabilité de fonctionnement pour une durée de vie donnée.

L'analyse PSA a été faite en respectant la norme ECSS-Q-ST-30-11C.

6.1 Les hypothèses de travail

Les hypothèses de travail sont présentées dans les sous-sections suivantes. Elles comprennent entre autres les informations relatives à la carte, aux composants utilisés, des tensions d'entrées, courants absorbés et puissances dissipées.

DERATING: intentional reduction in a parameter rating of a component in order to increase its useful life in terms of drift and reliability.

Les principaux paramètres de Derating :

- junction or case temperature at maximum operating conditions
- power (rating, dissipation)
- voltage
- current

Pour la puissance, un Derating de 25% a été utilisé car la norme ne précisait pas cette valeur pour tous les composants.

6.1.1 Liste des documents d'entrées : schéma et liste de pièces

La liste des composants est disponible dans la BOM (*Bill of material*) du projet. Pour les composants les plus importants, les Datasheets sont disponibles dans le répertoire du même nom.

La schématique a été présentée dans une des sections précédentes.

6.1.2 Températures considérées : carte, composants

Une valeur d'environ 110 °C a été prise pour tous les composants présents sur la carte, cela nous permet d'avoir des valeurs de Derating d'environ 50% en puissance pour la plupart des composants.

6.1.3 Puissances considérées : max, typ

Pour un bon fonctionnement, la puissance de dissipation des composants a été surdimensionnée pour avoir une bonne Derating et pas trop de stress sur ceux-ci.

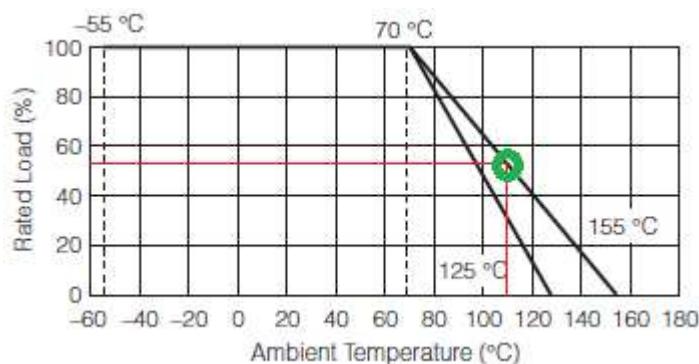
6.1.4 Liste des documents applicables considérés pour les valeurs de Derating

Les Datasheets des composants, notamment les résistances et les condensateurs, mais surtout, la norme ECSS-Q-ST-30-11C sont utilisés pour faire cette analyse.

6.2 La synthèse de l'analyse

Exemple avec la résistance shunt :

Cette résistance fonctionne jusqu'à une température de 155°C, nous fixons un Derating de 75% pour avoir une température d'environ 110°C (116.25°C). Dans le Datasheet, un schéma montre la courbe de Derating de la puissance en fonction de la température. Pour un composant fonctionnant jusqu'à 155 °C, une température de fonctionnement à 110 °C, ce qui induit un Derating de 50 % sur la puissance d'après le Datasheet. Dès lors la puissance est divisée par 2, soit 250mW. D'après les calculs de puissances dissipées du budget (242mW), il reste encore une marge de 8mW. La figure suivante montre le Derating de la puissance de la shunt en fonction de la température :



Certains composants ne passent pas le test de stress. Ces composants ne seront normalement pas montés lors du lancement du CanSat dans l'espace. Il n'est donc pas nécessaire de corriger ce problème, ils ne servent que pour la programmation et le debug de la carte.

6.3 Dimensionnement du filtre passe-bas

Le dimensionnement du filtre passe bas est fait à l'entrée du convertisseur analogique digital pour que la tension ne varie pas plus que $\frac{1}{2}$ LSB pendant la durée de conversion.

Le « ADS1282 » possède un Programmable gain amplifié intégré au composant. Il est possible de faire une division jusqu'à 64 et d'un FIR.

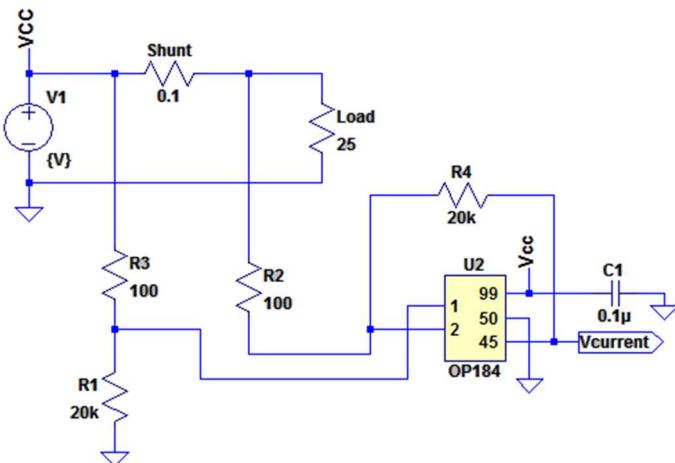
Une capacité de 10nF doit être connectée à la pin CAPP et CAPN de l'ADC pour filtrer les glitches du modulateur.

Avec la formule suivante : $f_p = 1/(6.3 \cdot 600 \cdot C)$ pour une fréquence de 25kHz, cela donne un condensateur d'environ 10nF. Cette fréquence dépend fortement de la variation de la tension d'entrée. Pour notre mesure, la tension et le courant sont relativement stables. Il ne devrait pas y avoir beaucoup de variations de tension et courant au fil du temps.

7 Worst Case Analysis (WCA)

Dossier de référence : /WCA

Nous avons choisi d'analyser le bloc de notre circuit qui nous semble le plus critique, celui de la mesure de courant.



$$Gain = R_{shunt} * \frac{R_1}{R_3}$$

7.1 Détermination des paramètres à considérer

Les paramètres à considérer pour les résistances sont donnés par les slides du cours :

- Résistances :
 - Température : Dérive et valeur aléatoire de la résistance
 - Humidité : Dérive de la résistance et composition du carbone
 - Durée de vie, alimenté : Dérive et valeur aléatoire de la résistance
 - Durée de vie, non alimenté : Dérive et valeur aléatoire de la résistance
 - Vide d'aire : Dérive de la résistance
 - Mécanique : Dérive et valeur aléatoire de la résistance

7.2 Détermination des performances à analyser

Nous allons analyser le gain du circuit avec la formule donnée plus haut.

7.3 Analyse Monte Carlo

Le gain du circuit est défini par les résistances. Selon les paramètres à considérer que nous avons trouvés, il faut tenir compte de la valeur aléatoire des résistances dues aux tolérances données par les fabricants.

Toutes nos résistances ont une variation de $\pm 1\%$. Nous avons donc entré les formules suivantes dans l'analyse :

Rsense	R3	R1	Gain
=0.1*(1+0.02*(ALEA()-0.5))	=100*(1+0.02*(ALEA()-0.5))	=20000*(1+0.02*(ALEA()-0.5))	= Rsense * R1/ R3

La valeur de gain espérée est de 20.

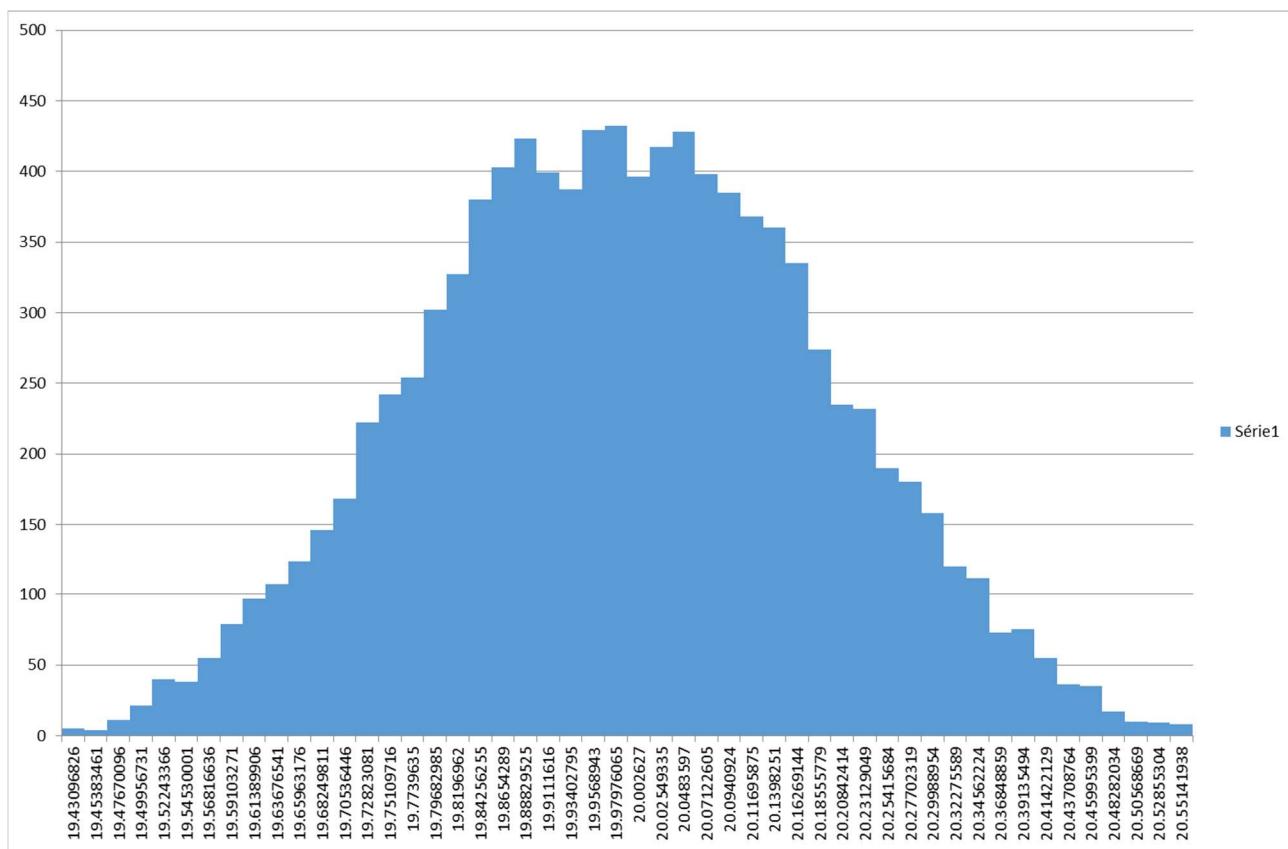
! Nous n'avons pas fait l'analyse avec PSPICE, mais avec le fichier Excel fourni. On pourrait faire l'analyse avec PSPICE ou LTSpice, cela nous fournirait les mêmes résultats. L'avantage d'utiliser LTSpice est que l'on n'a pas besoin d'entrer les formules, il suffit de donner les précisions des résistances. Cela peut être très utile avec un grand circuit à simuler. Le désavantage par rapport au fichier Excel est que les résultats ne sont pas donnés sous forme graphique. Comme les équations à simuler sont relativement simples, nous avons choisi d'utiliser la feuille Excel.

7.4 Evaluation des résultats

En faisant 10'000 répétitions, l'analyse fourni les résultats suivants :

	Rsense	R3	R1	Gain
10000	0.099946185	99.95695493	20043.73163	20.04157194
Mean	0.099994503	100.001949	20000.19244	19.99936833
Standard error	5.75759E-06	0.005773232	1.155844251	0.001997404
Median	0.099995216	99.99724678	20000.3514	19.99759769
Standard deviation	0.000575759	0.577323173	115.5844251	0.199740416
Variance	3.31498E-07	0.333302046	13359.75934	0.039896234
Skewness	-0.003398894	-0.005985113	0.003550935	0.024562599
Kurtosis	1.795861164	1.80703059	1.803513243	2.610631784

On peut constater que sur les 10'000 répétitions, on obtient un gain très proche des 20 espérés. La figure ci-dessous présente les résultats de manière graphique.



✓ Le résultat de l'analyse Monte Carlo valide le choix des résistances pour le gain de l'amplificateur. Leur imprécision donne une plage 3σ de : $19.43 \leq \text{Gain} \leq 20.55$, ce qui correspond à $\pm 2.75\%$ du gain espéré pour le pire des cas. Cela nous convient, on est en dessous des 3% et le graphique est correctement centré autour de 20.

8 Analyse de fiabilité (FRA)

Dossier de référence : /FRA

Ce test permet de mettre en place de la redondance froide pour que le système, une fois lancé, puisse être redémarré dans une autre configuration si des problèmes avec le système de base sont détectés.

Dans un premier temps, un schéma est mis en place pour pallier à d'éventuels problèmes de survie du bloc fonctionnelle. C'est la redondance.

Ensuite, une analyse FIT permet de comprendre combien de pannes sont possibles par millions d'heures d'utilisations. Elle fait état de comparaison avec le système à redondance froide.

Les valeurs inscrites dans le tableau de FIT sont issues de la norme MIL-HDBK-217F.

8.1 Circuit de mesure à redondance froide

Le circuit proposé ci-dessous pour une redondance froide couvre les parties suivantes : le bloc alimentation, le bloc de mesure avec la résistance shunt et la conversion analogique digitale.

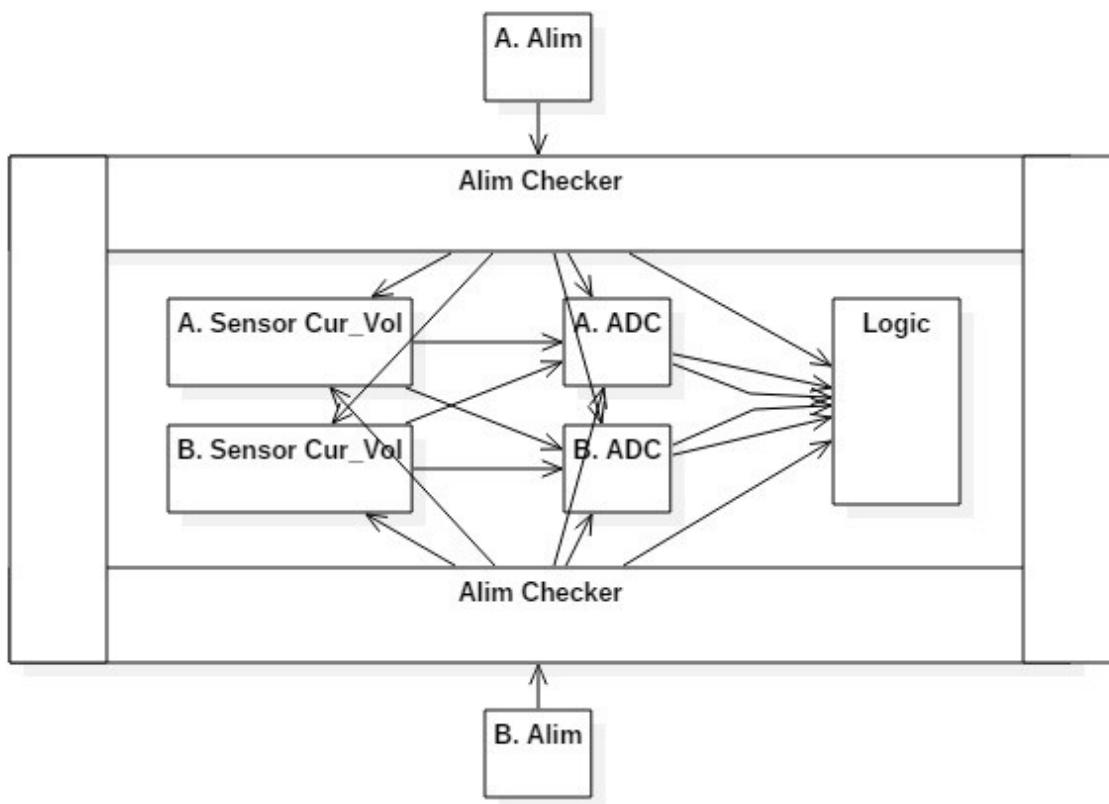


Figure 1 Redondance froide

Les signaux logiques sur la FPGA sont dupliqués sur plusieurs entrées pour que le signal puisse être récupéré si la FPGA n'est plus opérationnelle. Ex : paire différentielle, pour avoir une référence à double.

Pour que la redondance fonctionne dans ce cas de figure, il faut intégrer un circuit supplémentaire pour faire le switching au niveau de l'alimentation et de la mesure. Si un problème surgit sur la mesure, il est fort probable que le reste du circuit ne puisse plus être alimenté du fait que la shunt monté sur le bloc de mesure peut couper l'alimentation du reste de la carte. Pour ce faire, un bloc analogique est monté entre les alimentations du reste du circuit pour garantir une alimentation stable et continue du reste de la carte.

La partie « Alim Checker » sur la figure ci-dessus permet de sélectionner laquelle des alimentations est activée.

Une autre solution est de mettre en place une redondance chaude pour les deux alimentations ce qui permettrait d'avoir les deux alimentations activées et de maintenir toujours une tension d'alimentation de la carte. Il est tout de même nécessaire de mettre en place des protections au niveau des alimentations pour ne pas avoir de court-circuit avec la redondance quand celle-ci est activée.

8.2 Questions

- Les circuits non alimentés supportent la présence de signaux à leurs entrées ? :

Les circuits qui ne sont pas alimentés supportent la présence de signaux aux entrées, car les composants présents sont munis d'une impédance d'entrée très grande et possèdent une compensation d'entrée grâce à la contre-réaction, ce qui stabilise les signaux.

- L'état du système lors de la première mise sous tension ? :

Pour le mode avec « Alim Checker », l'état peut être une alimentation ou l'autre. Ensuite la partie logique s'occupera de sélectionner un étage de conversion et mesure ou bien de mixer les deux.

Pour la redondance chaude, les alimentations ont besoin d'être activées en même temps pour garantir qu'une alimentation au moins fonctionne et fournit une tension au reste de la carte.

8.3 Nombre de pannes par milliard d'heures (Failure in Time, FIT)

Ce premier tableau liste les composants de notre carte sans redondance :

Avec cette configuration, le taux de pannes est chiffré à 0.20 pannes par million d'heures de fonctionnement.

#	QTY	REFDES	DEVICE	VALUE	λ_b	C1	C2	π_T	π_E	π_Q	π_L	π_S	π_R	π_C	π_{CV}	λ_p	λ_e
1	20	R1...R20	Ceramic - 0805		0.00011			0.5	1			1			0.000055	0.0011	
2	6		Ceramic - 0603		0.00089			0.5	1					1	0.000445	0.00267	
3	9		Ceramic - 0805		0.00089			0.5	1					1	0.000445	0.004005	
4	1		Tantal - A case		0.022			0.4	1					1	0.0088	0.0088	
5	2		Tantal - B case		0.022			0.4	1					1	0.0088	0.0176	
6	1	U6	OP184	Ampli op		0.02	0.019	0.1	0.5	1	1				0.0115	0.0115	
7	1	U5	Convertisseur A/D ADS1282-HT	Analogique to digital			0.6	0.056	0.1	0.5	1	1			0.088	0.088	
8	1	U3	MAX865	Doubleur de tension		0.04	0.026	0.1	0.5	1	1				0.017	0.017	
9	1	U4	MAX6350	Régulateur de tension 5V		0.04	0.026	0.1	0.5	1	1				0.017	0.017	
10	1	U2	FTDI FT232RL	USB to UART		0.02	0.019	0.1	0.5	1	1				0.0115	0.0115	
11	1	U1	AP7361	Convertisseur DC/DC Adj.1V to 5V		0.04	0.026	0.1	0.5	1	1				0.017	0.017	
12	2	D1,D2	LED verte	Indicateur visuel	0.0038			0.1	0.5	1					0.0019	0.0038	

λ_{Ptot} 0.199975

Comparaison de FIT avec le circuit redondant :

Ce deuxième tableau fait état d'une électronique munie d'un système à redondance. Il est dès lors possible de réduire considérablement le taux de pannes par heures à 0.12.

12 Avril 2016

#	QTY	REFDES	DEVICE	VALUE	λ_b	C1	C2	π_T	π_E	π_Q	π_L	π_S	π_R	π_C	π_{CV}	λ_p	λ_e
1	20	R1...20	Ceramic - 0805		0.00011				0.5	1		1			0.000055	0.0011	
2	6		Ceramic - 0603		0.00089				0.5	1				1	0.000445	0.00267	
3	9		Ceramic - 0805		0.00089				0.5	1				1	0.000445	0.004005	
4	1		Tantal - A case		0.022				0.4	1				1	0.0088	0.0088	
5	2		Tantal - B case		0.022				0.4	1				1	0.0088	0.0176	
6	2	U6	OP184	Ampli op		0.02	0.019	0.1	0.5	1	1				0.0115	0.023	
7	2	U5	Convertisseur A/D ADS1282-HT	Analogique to digital			0.6	0.056	0.1	0.5	1	1			0.088	0.176	
8	2	U3	MAX865	Doubleur de tension			0.04	0.026	0.1	0.5	1	1			0.017	0.034	
9	2	U4	MAX6350	Régulateur de tension 5V			0.04	0.026	0.1	0.5	1	1			0.017	0.034	
10	1	U2	FTDI FT232RL	USB to UART			0.02	0.019	0.1	0.5	1	1			0.0115	0.0115	
11	2	U1	AP7361	Convertisseur DC/DC Adj.1V to 5V			0.04	0.026	0.1	0.5	1	1			0.017	0.034	
12	2	D1,D2	LED verte	Indicateur visuel	0.0038				0.1	0.5	1				0.0019	0.0038	

$$\begin{aligned} \lambda_{\text{Para}} &= 0.00575 & 0.044 & 0.0085 & 0.0085 & 0.0085 \\ \lambda_{\text{Ptot}} & & & & & \end{aligned}$$

0.124725

9 Systèmes numériques

Dossier de référence : /VHDL/CanSat

Nous avons reçu un canevas de base du cansat contenant une représentation de notre circuit de conversion A/D.

Nous avons commencé par essayer de simuler le circuit de base. Pour cela, nous avons utilisé le logiciel Questasim dont la licence nous a été fournie dans un autre cours.



Des captures d'écran de toutes les simulations sont présentes dans ce document au cas où le code ne serait pas simulable avec un autre logiciel.

L'exécutable compilant le projet, *modelsim.bat*, a dû être modifié pour utiliser questasim et non modelsim.

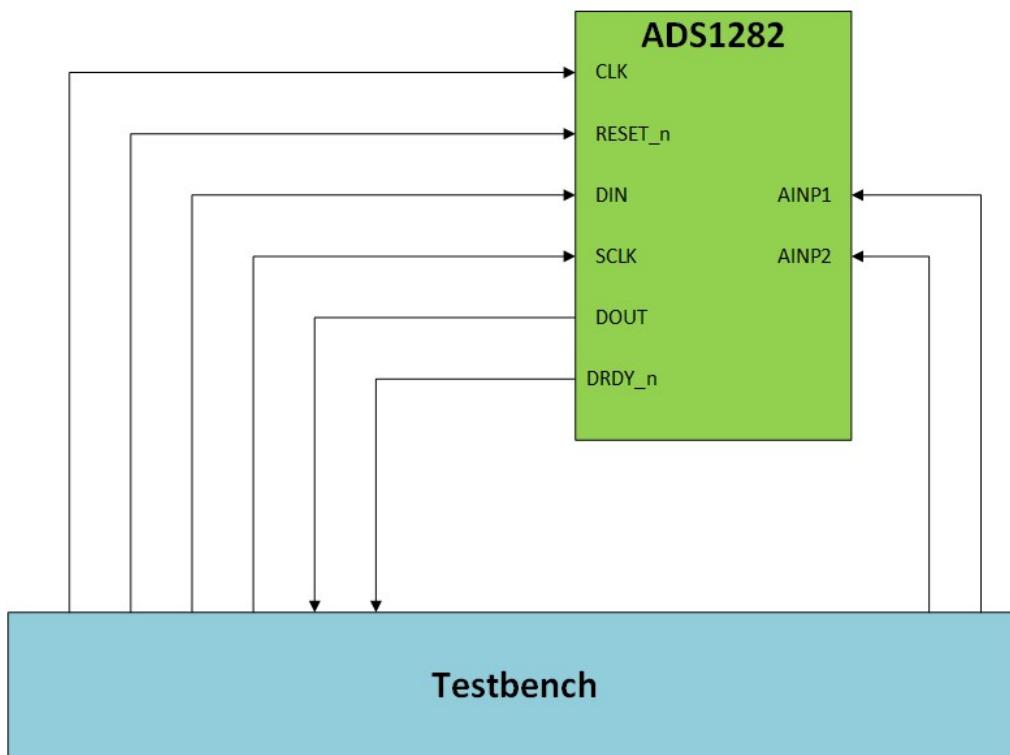
10 Périphérique de contrôle

10.1 Convertisseur A/D

Dossier de référence : /VHDL/CanSat

10.1.1 Modification de l'entité

Une fois le circuit base compilé et simulé, nous avons modifié l'entité représentant le convertisseur ADS1282. Elle a été modifiée pour correspondre à l'image ci-dessous.



Certains signaux tels que SYNC ou PWD_n ont été enlevés pour refléter au mieux la réalité de notre circuit. Dans notre schématique, nous ne disposons que de six pins pour communiquer entre le convertisseur et la FPGA. Nous n'avons donc pas pu relier absolument tous les signaux de contrôle du convertisseur. Nous avons donc conservé uniquement les indispensables.

10.1.2 Adaptation de convertisseur pour une lecture par commande

Le fichier *ads1282_sim.vhd* contenant une modélisation du comportement du convertisseur A/D a ensuite été modifié. Le comportement implémenté ne correspond pas à celui que nous voulons. On veut utiliser les deux canaux du convertisseur pour obtenir en alternance une mesure du courant et une autre de la tension de notre système.

Par défaut, le convertisseur est en mode « *Read Data Continuous* » sur le canal 1. Nous voulons pouvoir modifier la configuration pour lire les deux canaux. Pour cela, il faut utiliser les commandes suivantes :

1. Stopper la lecture en continu⁸

COMMAND	TYPE	DESCRIPTION	1st COMMAND BYTE ⁽¹⁾⁽²⁾	2nd COMMAND BYTE ⁽³⁾
SDATAC	Control	Stop read data continuous	0001 0001 (11h)	

2. Configurer le canal à utiliser :

⁸ Source des images : Datasheet du ADS1282

COMMAND	TYPE	DESCRIPTION	1st COMMAND BYTE ⁽¹⁾⁽²⁾	2nd COMMAND BYTE ⁽³⁾
WREG	Register	Write <i>nnnn</i> register(s) at address <i>rrrr</i>	010r <i>rrrr</i> (40h + 000r <i>rrrr</i>)	000n <i>nnnn</i> (00h + n <i>nnnn</i>)

Pour cela, il faut utiliser la commande WREG pour écrire le registre CONFIG1.

ADDRESS	REGISTER	RESET VALUE	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
02h	CONFIG1	08h	0	MUX2	MUX1	MUX0	CHOP	PGA2	PGA1	PGA0

8.6.3.2 Configuration Register 1

Figure 64. CONFIG1: Configuration Register 1 (Address 02h)

7	6	5	4	3	2	1	0
RSVD	MUX2	MUX1	MUX0	CHOP	PGA2	PGA1	PGA0
0							

Reset value = 08h

Table 14. Configuration Register 1 Field Descriptions

Bit	Field	Type	Reset	Description
7	Reserved			Always write '0'
6:4	MUX[2:0]			MUX Select 000: AINP1 and AINN1 (default) 001: AINP2 and AINN2 010: Internal short via 400Ω 011: AINP1 and AINN1 connected to AINP2 and AINN2 100: External short to AINN2
3	CHOP			PGA Chopping Enable 0: PGA chopping disabled 1: PGA chopping enabled (default)
2:0	PGA[2:0]			PGA Gain Select 000: G = 1 (default) 001: G = 2 010: G = 4 011: G = 8 100: G = 16 101: G = 32 110: G = 64

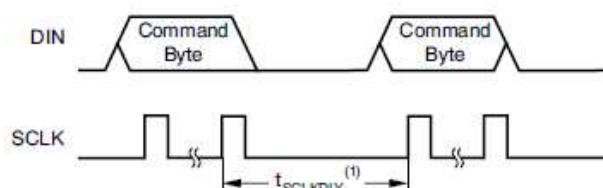
Donc pour configurer la conversion sur le canal 1, il faut envoyer la suite de bytes suivante : 0h420008. Et pour le canal 2, la séquence 0h420018.

3. Lire par commande :

COMMAND	TYPE	DESCRIPTION	1st COMMAND BYTE ⁽¹⁾⁽²⁾	2nd COMMAND BYTE ⁽³⁾
RDATA	Data	Read data by command ⁽⁴⁾	0001 0010 (12h)	

10.1.3 Timing à respecter

Pour envoyer plusieurs commandes à la suite, il faut respecter 24 coups de clock entre deux commandes :



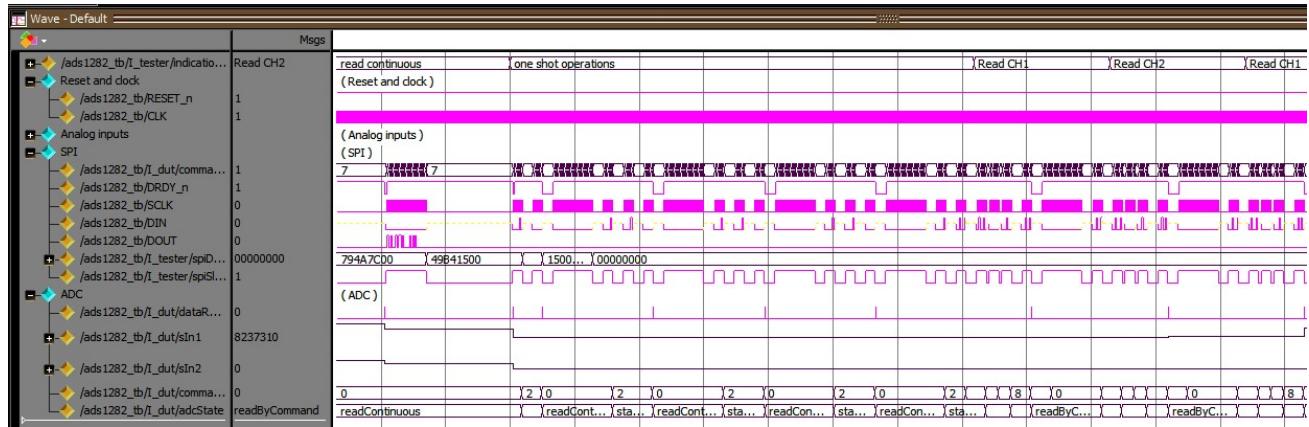
(1) $t_{SCLKDLY} = 24/f_{CLK}$ (min).

Figure 57. Consecutive Commands

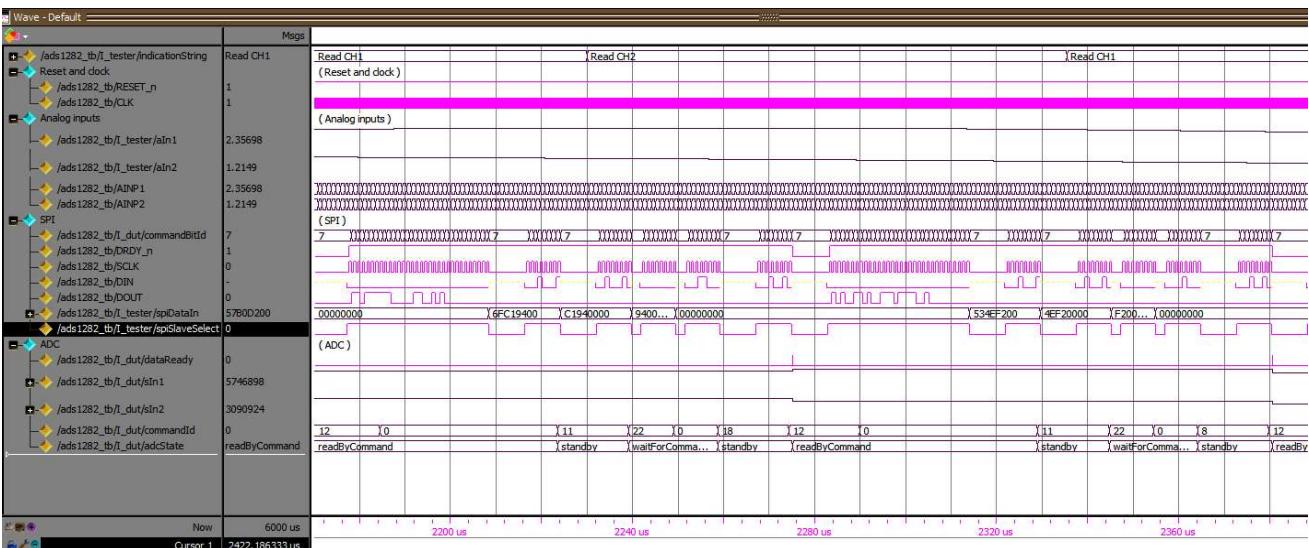
10.1.4 Simulation de l'ADS1282

La représentation de notre convertisseur a donc été modifiée pour qu'il reconnaisse les trois commandes décrites précédemment. Nous avons fait en sorte de pouvoir envoyer une commande contenant plusieurs bytes pour la configuration du canal. Le canal à lire est modifié lors de la configuration du registre CONFIG1.

Le fichier ads1282_tester.vhd contenant le testbench du convertisseur ensuite été modifié. Nous l'avons adapté pour qu'il envoie les commandes précédemment implémentées et ainsi tester le bon fonctionnement du système. La simulation suivante est obtenue :



Le fonctionnement de base du circuit n'a pas été altéré. On peut toujours faire une lecture en continu ainsi qu'un « *one shot operation* ». Nous avons complété le banc de test pour lire le canal 1 suivi du 2 une vingtaine de fois.



✓ En regardant de plus près la lecture des canaux, on observe que l'on a bien une valeur qui est envoyé sur la sortie DOUT lors d'une demande de lecture. Nous pouvons donc à ce stade valider la bonne implémentation des commandes de contrôle ainsi que la lecture par commande du convertisseur AD.

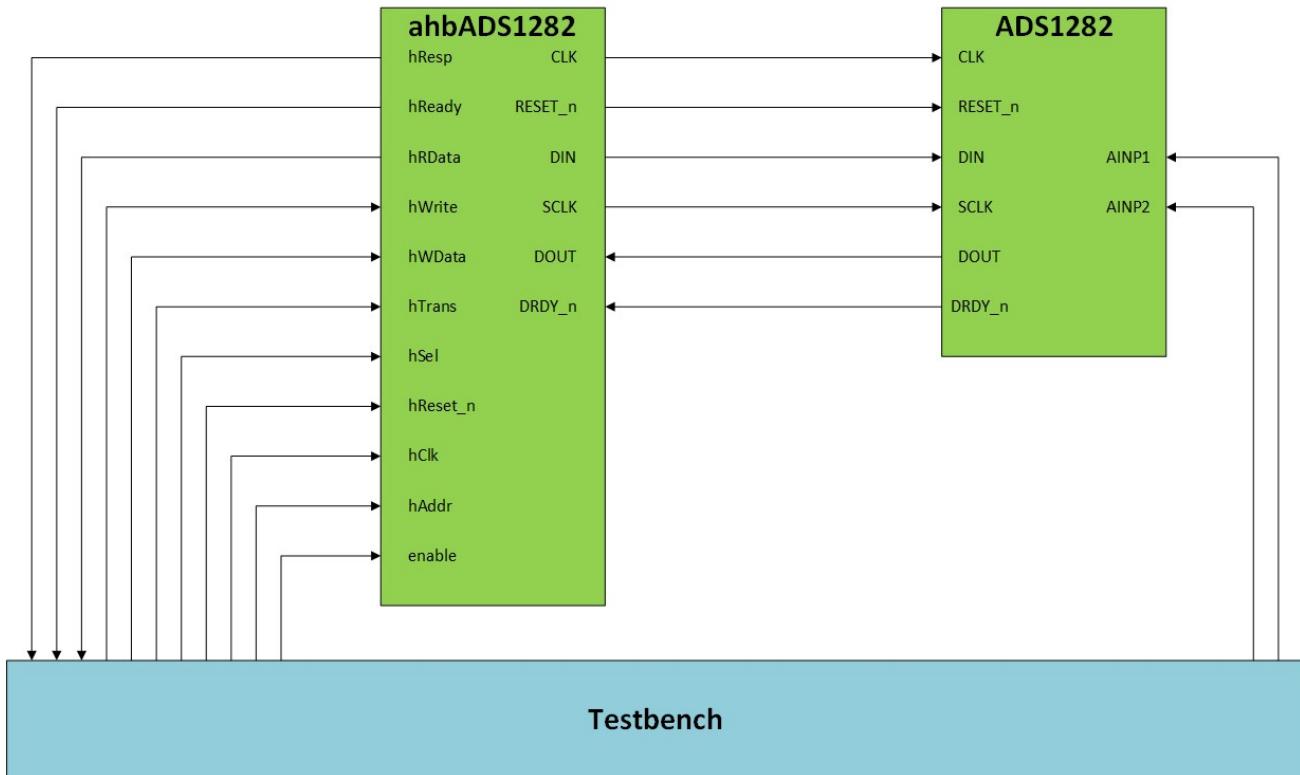
10.2 Périphérique AMBA

Dossier de référence : /VHDL/CanSat

10.2.1 Modification de l'entité

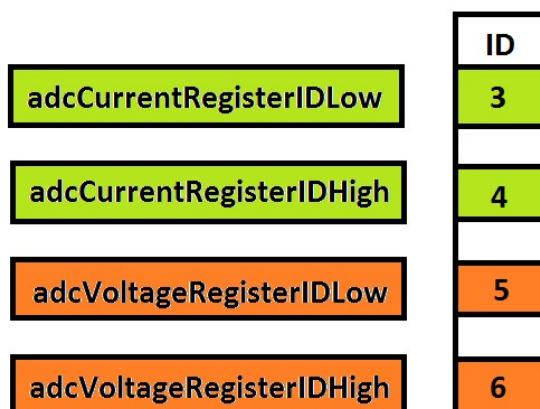
Nous avons également reçu une simulation de convertisseur avec un composant nommé ahbAds1282 qui permet au bus AMBA d'envoyer des instructions pour piloter le convertisseur.

L'entité a été modifiée pour correspondre à l'image ci-dessous :



10.2.2 Modification du contrôleur

Nous avons ensuite modifié le contenu du contrôleur, *ahbAds1281_RTL.vhd*. Nous avons ajouté quatre registres pour permettre de récupérer la valeur du courant et de la tension. Chacune de ces deux valeurs est partagée dans un registre contenant son MSB et l'autre son LSB.



Ces registres sont accessibles uniquement en lecture. Chaque fois que l'entrée enable est active, le circuit fait une lecture d'un canal du convertisseur et stocke sa valeur dans le registre correspondant, voltage pour le canal 1 et current pour le canal 2. Le canal à lire est alterné après chaque lecture.

10.3 Machine d'état du contrôleur ADC

Dossier de référence : /VHDL/MachineEtat

La machine d'état comporte maintenant huit états. Dès que l'entrée enable est active, on va configurer le convertisseur A/D et faire une lecture simple sur un de ses canaux.

10.4 Codage de la machine d'états

Dossier de référence : /VHDL/CanSat

10.4.1 Signaux de contrôle

Comme le code n'était pas pensé pour envoyer plusieurs commandes à la suite, ni des commandes de plusieurs bytes, trois processus ont été implémentés pour combler ce manque.

Celui-ci permet d'envoyer une commande de 3 bytes. Elle est faite sur mesure pour la configuration du registre CONFIG1, mais pourrait être améliorée en ajoutant un signal contenant la taille de la commande voulue.

```
adcLongCmd: process(adcState,reset, clock)
begin
  if reset = '1' then
    adcConfigured <= '0';
    adcConfigByteNbr <= 0;
  elsif rising_edge(clock) then
    adcConfigured <= '0';
    if adcState = sendConfigCH and adcSending = '0' and adcCommandWait = '0' and adcCmdTimeWait = '0' then
      adcConfigByteNbr <= adcConfigByteNbr+1;
      if adcConfigByteNbr >= 2 then
        adcConfigByteNbr <= 0;
        adcConfigured <= '1';
      end if;
    end if;
  end if;
end process adcLongCmd;
```

Ce deuxième processus attend que la commande actuelle soit envoyée en levant un signal empêchant la machine d'état d'en envoyer d'autres.

```
adcWaitCmdSend: process(reset,clock,adcSendCommand)
begin
  if reset = '1' then
    adcCommandWait <='0';
  elsif rising_edge(adcSendCommand) then
    adcCommandWait <='1';
  elsif rising_edge(clock) then
    adcCommandWait <='0';
    if adcSendCommand = '1' and adcSending = '0' and adcCounterEnable = '0' then
      adcCommandWait <='1';
    end if;
  end if;
end process;
```

Finalement, celui-ci attend les 24 coups d'horloge entre les commandes :

```
adcCmdTimeWait24: process(reset, modulatorClock, adcSending, adcSendCommand, adcState, adcConfigByteNbr)
begin
  if reset = '1' then
    adcTimecounter <= 0;
    adcCmdTimeWait <= '0';
    adcCounterEnable <= '0';
  elsif adcSendCommand = '1' then
    if adcState /= sendConfigCH or (adcState=sendConfigCH and adcConfigByteNbr >= 2) then
```

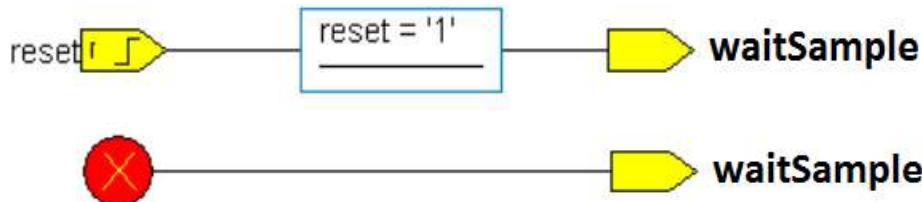
```

if rising_edgeadcSending) then
    adcCmdTimeWait <= '1';
elsif falling_edgeadcSending) then
    adcTimecounter <= 0;
    adcCounterEnable <= '1';
    adcCmdTimeWait <= '1';
elsif rising_edge(modulatorClock) and adcCounterEnable = '1' then
    adcCmdTimeWait <= '1';
    adcTimecounter <= adcTimecounter + 1;
    if adcTimecounter >= 24 then
        adcTimecounter <= 0;
        adcCounterEnable <= '0';
        adcCmdTimeWait <= '0';
    end if;
end if;
end if;
end process adcCmdTimeWait24;

```

10.4.2 Fiabilisation

Pour éviter d'avoir des incertitudes avec les états, nous avons fait en sorte d'être dans l'état waitSample lors d'un reset ou lorsque l'on est dans un état non défini.

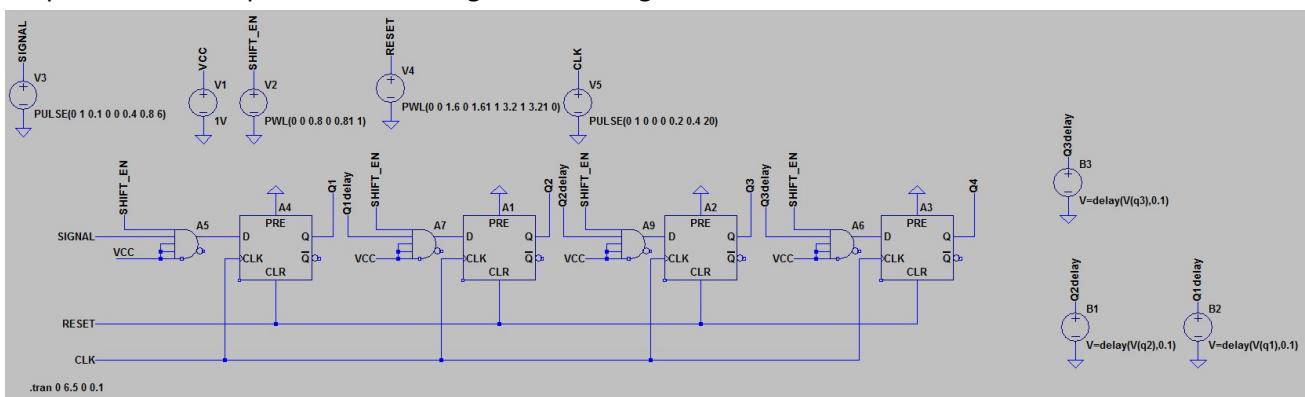


11 Effets des radiations

11.1 Registre à décalage pour réception d'un signal série

Dossier de référence : /EffetRadiations/RegistreADecalage

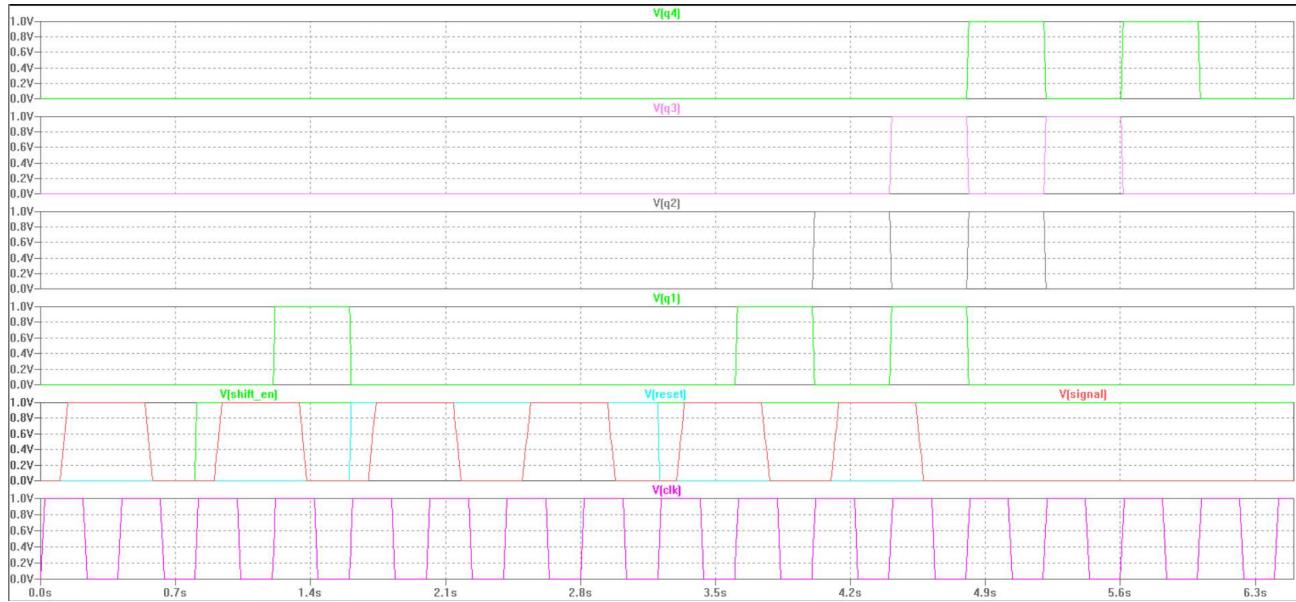
LTS spice a été utilisé pour simuler un registre à décalage 4 bits. Voici le circuit de base :



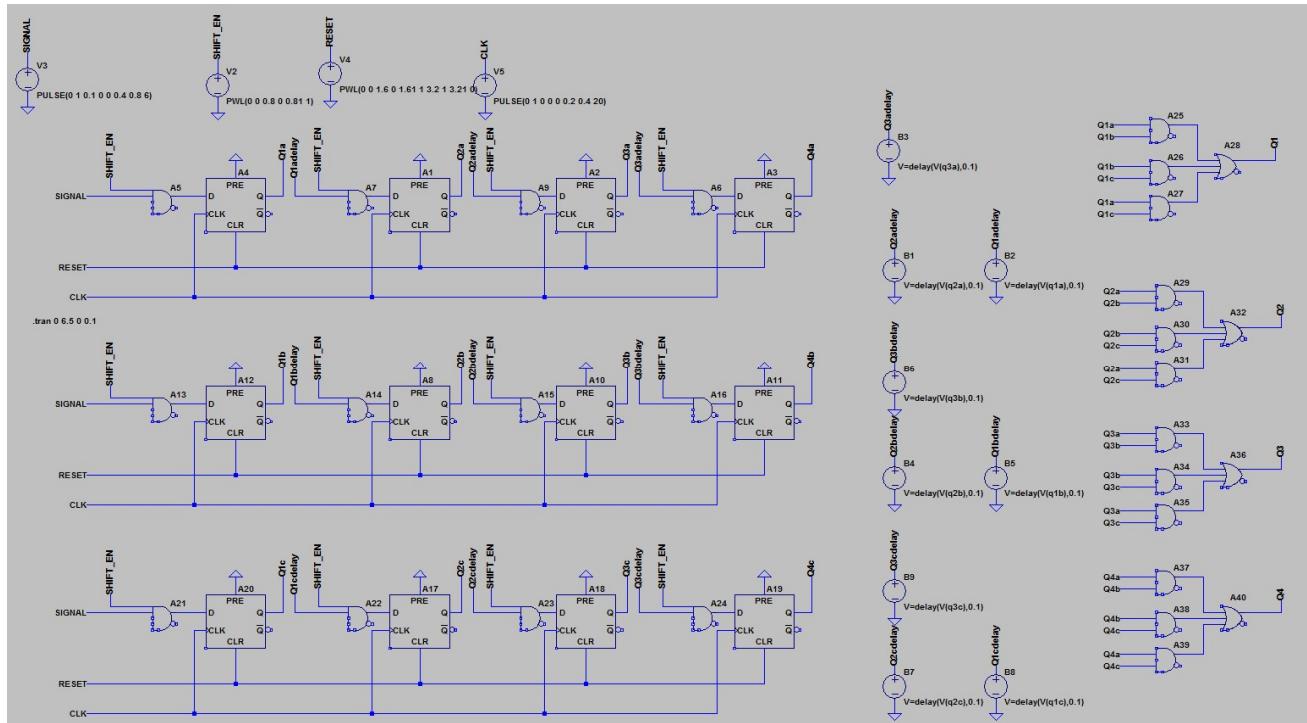
Un problème a été constaté avec le composant de la bascule D. Il n'y a pas de temps de propagation entre l'entrée et la sortie, cela induit que le signal est propagé en un seul coup de clock jusqu'à la sortie Q4.

Pour résoudre ce problème, nous avons ajouté des délais de propagation à l'aide de sources BV.

Voici une image validant le fonctionnement de notre registre à décalage. On voit que le signal n'est pas propagé en cas de reset ou lorsque le signal SHIFT_ENABLE n'est pas actif.



Le circuit a ensuite été dupliqué trois fois. Les signaux de contrôles sont les mêmes pour chacun des circuits. Les sorties Q1, Q2 et Q3 de chaque circuit passent par une logique de vote à la majorité séparée pour définir le signal de sortie final.

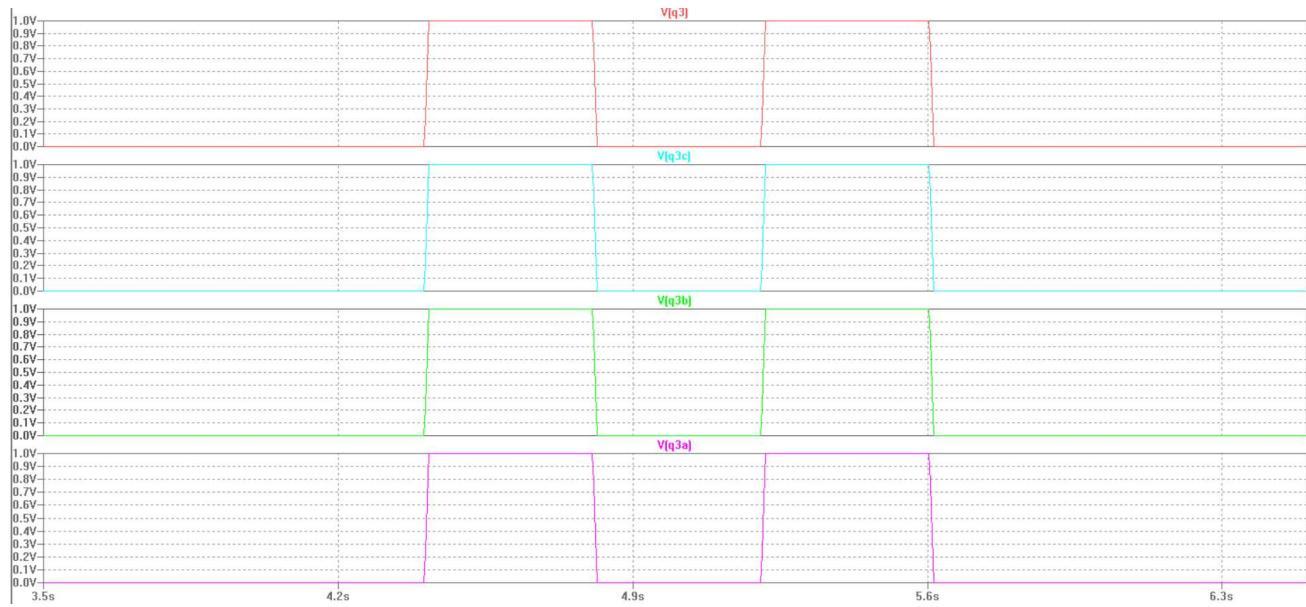


Voici la table de Karnaugh utilisée pour définir le circuit logique de vote à la majorité.

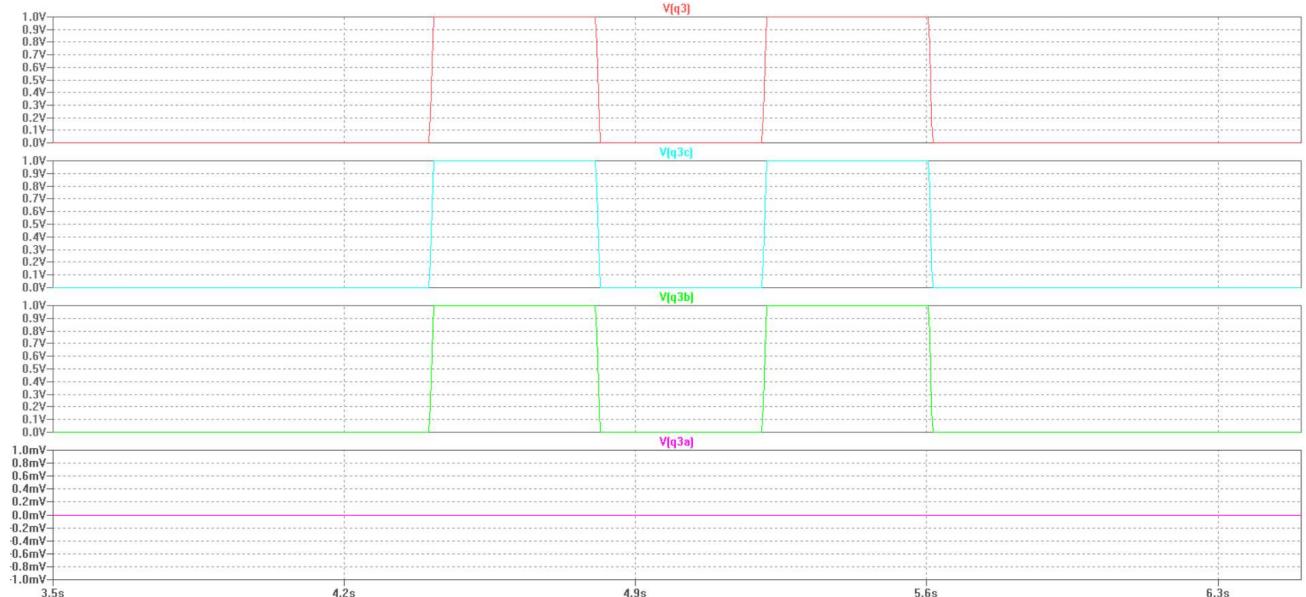
		Q1ab			
		00	01	11	10
Q1c	0	0	0	1	0
	1	0	1	1	1

$$\text{Équation: } Q1a \cdot Q1b + Q1b \cdot Q1C + Q1a \cdot Q1c$$

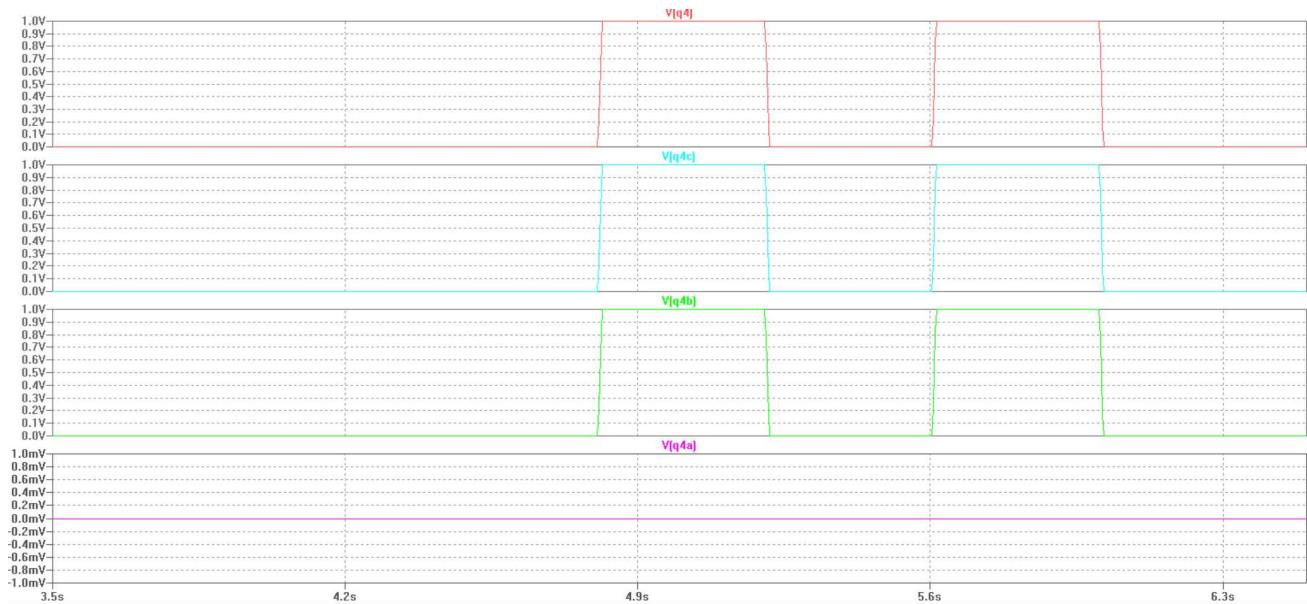
Voici le résultat de la simulation sur la sortie Q3 lorsque tous les composants fonctionnent correctement



Si on simule une panne sur l'entrée Q3a, on peut constater que notre circuit fournit quand même la bonne valeur de sortie sur Q3.



Bien entendu, une panne de la bascule Q3 implique que la bascule Q4 est également affectée. La valeur de Q4 est également correctement corrigée.



- ✓ Avec la triple redondance, on arrive à conserver des signaux corrects même si un des trois circuits ne fonctionne plus du tout.
- ✗ Par contre, dans le cas de ce registre à décalage, si une ou plusieurs bascules ne fonctionnent plus dans deux registres à décalage différents, les sorties ne sont plus correctes.

11.2 Machine d'état du contrôleur ADC avec contraintes SEU

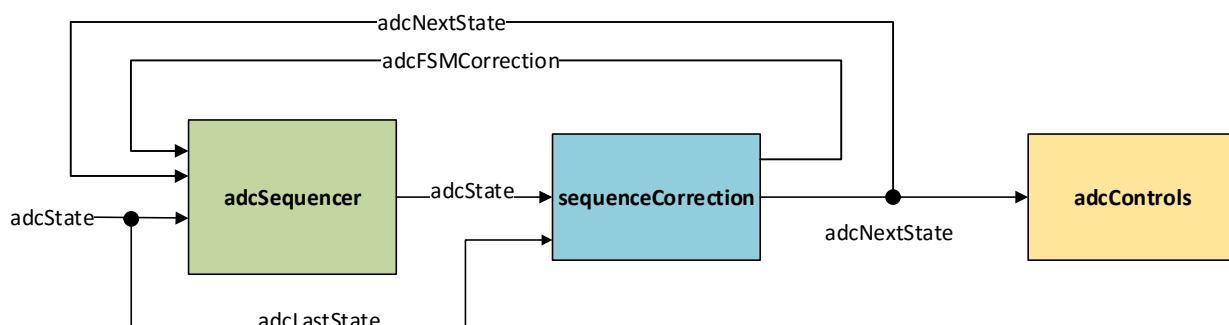
Dossier de référence : /VHDL/CanSat

Pour rendre notre machine d'état un peu plus robuste, les états ont été codés avec un codage 1 parmi m. Un seul bit est à 1 et se décale vers la gauche en fonction de la séquence suivante :

waitSample->sendSDATAC->sendConfigCH->sendReadByCmd->waitDataReady->startRead->waitRead->reading

```
constant waitSample          : fsm_stateType := "00000001";
constant sendSDATAC         : fsm_stateType := "00000010";
constant sendConfigCH        : fsm_stateType := "00000100";
constant sendReadByCmd       : fsm_stateType := "00001000";
constant waitDataReady       : fsm_stateType := "00010000";
constant startRead           : fsm_stateType := "00100000";
constant waitRead             : fsm_stateType := "01000000";
constant reading              : fsm_stateType := "10000000";
```

Comme nous avons huit états, il nous faut huit bits pour coder les états. La correction de la séquence en cas d'erreur est représentée dans la figure suivante :

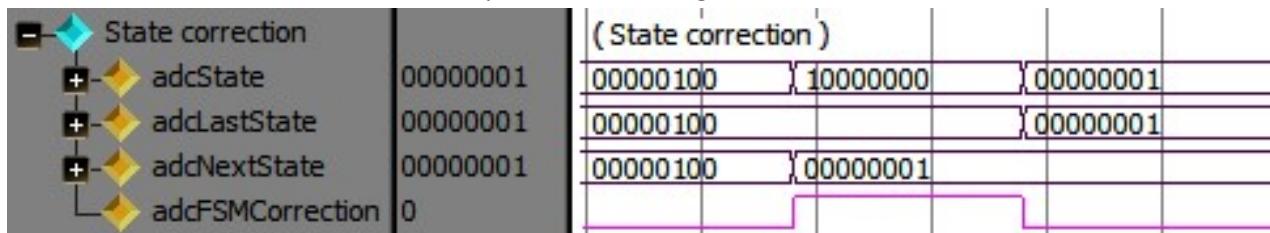


Le processus sequenceCorrection a été implémenté et contrôle que les états se suivent correctement. En cas d'erreur dans la séquence, la machine d'état est remise dans l'état waitSample.

Afin de valider le bon fonctionnement de notre correction d'état, nous avons volontairement introduit une erreur dans notre machine sur l'état sendConfigCH.

```
when sendConfigCH =>
  if adcConfigured = '1' then
    --adcState <= sendReadByCmd;      //Correct state "00001000";
    adcState <= "10000000";          // Wrong state, reading
  end if;
```

Voici le résultat de la simulation, on voit que l'état est corrigé correctement :



✓ La machine d'état se remet dans un état connu en cas de modification inopinée d'un bit du vecteur d'état. Lors de la détection de l'erreur indiquée par le signal adcFSMCorrection, le signal adcState pilotant la machine d'état est remis à l'état waitSample au cycle d'horloge suivant.

11.3 Décodeur de Hamming

Dossier de référence : [/VHDL/CanSat](#)

Comme notre système peut potentiellement être soumis à des radiations, notre machine d'état peut changer d'état sans que ce ne soit voulu, c'est pourquoi nous avons implémenté une correction d'état par codage de Hamming.

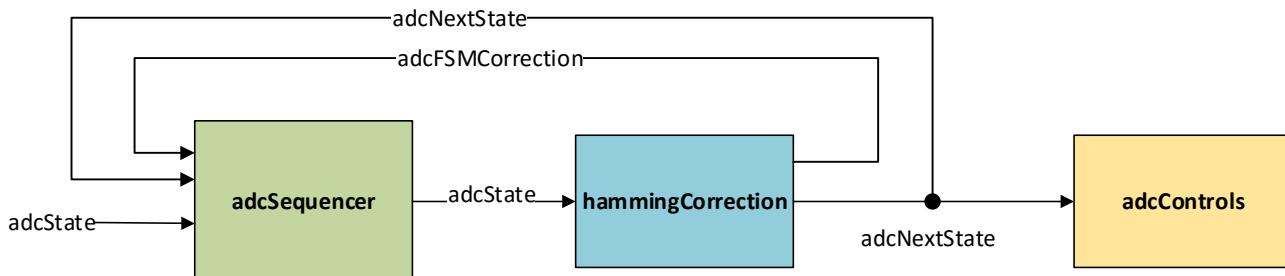
Nous avons huit bits de données. Selon le tableau ci-dessous, nous devons donc ajouter 4 bits de parité au codage des états. On dit que c'est un code (12,8). Les croix du tableau indiquent où appliquer les XOR pour calculer les bits de parité.

id	1	2	3	4	5	6	7	8	9	10	11	12
bit	p0	p1	d0	p2	d1	d2	d3	p3	d4	d5	d6	d7
parités	X		X		X		X		X		X	
		X	X			X	X			X	X	
				X	X	X	X					X
								X	X	X	X	X

Voici le codage des états en 1 parmi m avec en plus les bits de parité du codage de Hamming

```
constant waitSample      : hamming_stateType := "000000010011";
constant sendSDATAC     : hamming_stateType := "000000100101";
constant sendConfigCH    : hamming_stateType := "000001000110";
constant sendReadByCmd   : hamming_stateType := "000010000111";
constant waitDataReady   : hamming_stateType := "000100001001";
constant startRead       : hamming_stateType := "001000001010";
constant waitRead        : hamming_stateType := "010000001011";
constant reading         : hamming_stateType := "100000001100";
```

Nous avons ensuite implémenté un processus nommée *hammingCorrection* qui va analyser l'état courant et indiquer à l'aide du signal *adcFSMCorrection* si une erreur a été détectée sur l'état. Cela permet à la machine de recharger l'état correct. L'état correct est calculé et est donné sur le signal *adcNextState*. L'image ci-dessous présente le principe de notre codage de Hamming.



Afin de valider le bon fonctionnement de notre correction d'état, nous avons volontairement introduit une erreur dans notre machine sur l'état *sendConfigCH*.

```

when sendConfigCH =>
    if adcConfigured = '1' then
        --adcState <= sendReadByCmd;
        adcState <= "000010000111";           //Correct state "000010000111";
                                                //Wrong state, modify one bit
    end if;
  
```

Voici le résultat de la simulation. On voit que l'état est corrigé correctement, la machine d'état recharge l'état correct au flanc montant d'horloge suivant :



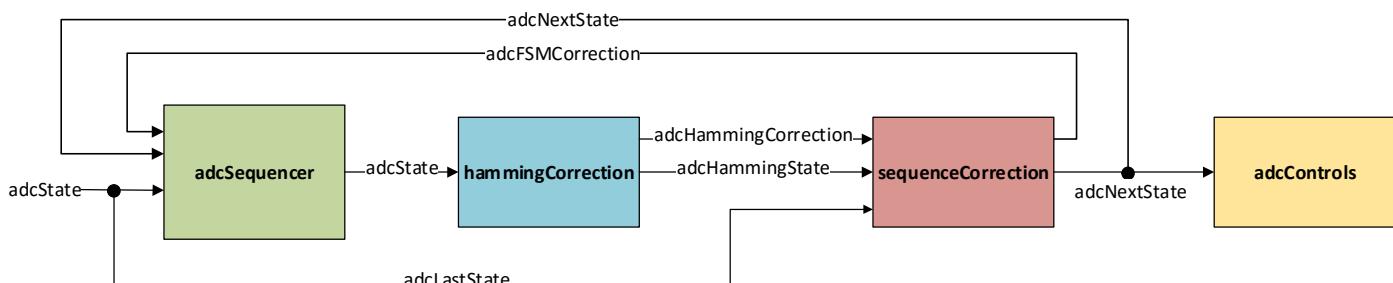
11.3.1 Réponse aux questions des slides

Avec le codage de Hamming, on peut corriger **une** erreur sur le vecteur d'état.

On peut détecter, dans notre cas, 12 erreurs sans pour autant les corriger. On peut donc dire qu'on peut détecter autant d'erreurs qu'il y a de bits dans le vecteur d'état (parité + donnée).

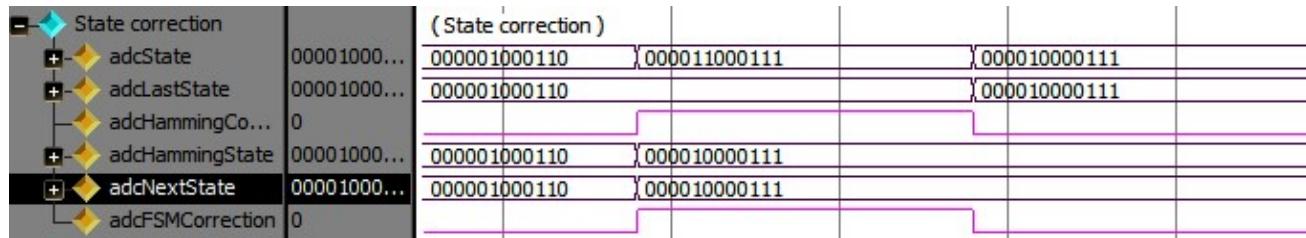
11.3.2 Mise en commun du codage de Hamming avec correction de séquence

Nous avons ensuite modifié les processus développés précédemment pour les mettre en commun. C'est-à-dire pouvoir corriger un bit du vecteur d'état et revenir dans un état connu si la séquence n'est pas bonne.

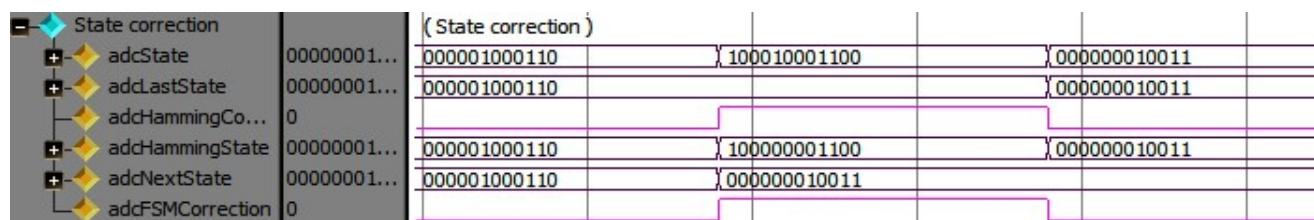


12 Avril 2016

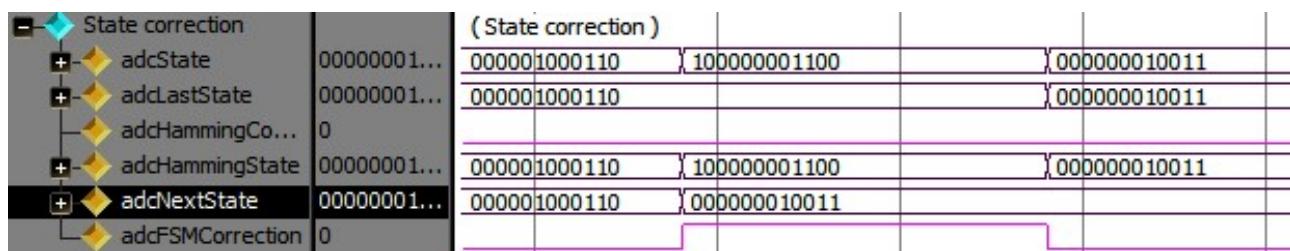
Cette simulation montre que l'état est correctement modifié lors qu'un bit du vecteur d'état change mais que la séquence est correctement suivie :



Cette simulation montre que l'on revient dans l'état waitSample lorsqu'un bit du vecteur d'état change mais que la séquence n'est pas correctement suivie :



On peut également encore faire le test si aucun bit du vecteur d'état ne change, mais que la séquence n'est pas suivie correctement.

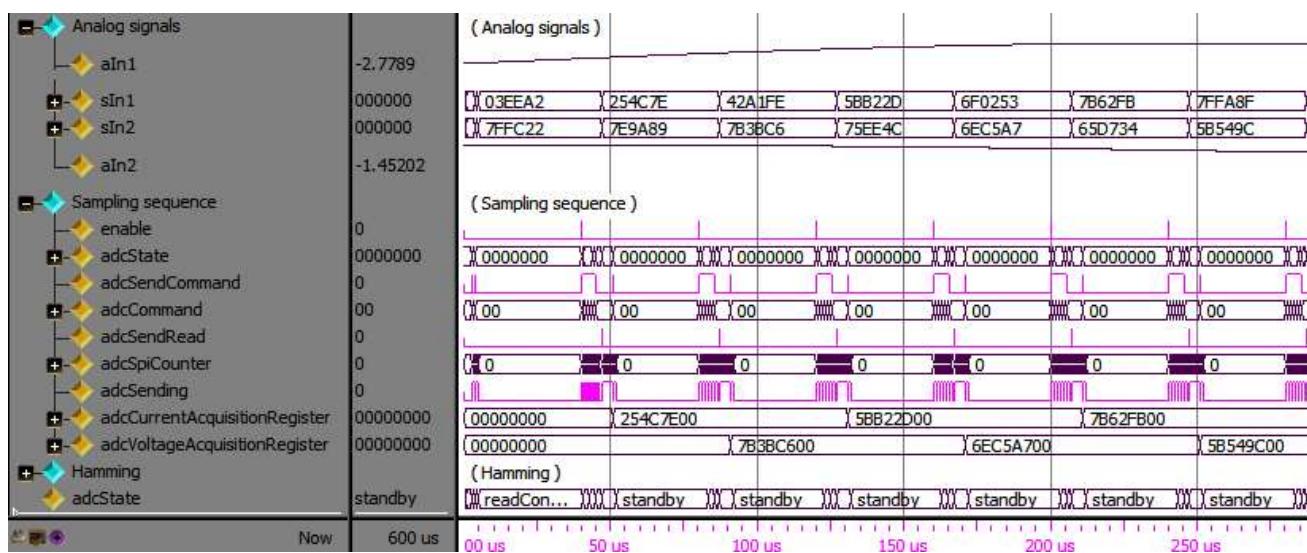


11.3.3 Transmission de l'erreur au registre de status

Dans l'exemple de code donné, un registre de status est implémenté. Nous avons fait en sorte que l'état du signal adcFSMCorrection se retrouve sur le bit 1 de ce registre pour pouvoir transmettre sur le bus AMBA s'il y a eu une erreur dans les états.

11.4 Simulation de l'ensemble du circuit

Avec cette simulation, on valide qu'à chaque enable, on arrive à récupérer correctement la valeur échantillonnée par le convertisseur A/D. On voit dans l'image que l'on lit correctement en alternance la valeur de sIn1 et sIn2 correspondant aux deux canaux du convertisseur. Les valeurs sont correctement récupérées dans les deux registres correspondants *adcCurrentAcquisitionRegister* et *adcVoltageAcquisitionRegister*.



À ce stade on a donc :

- ✓ Une machine d'état robuste dont l'état peut être corrigé en cas de perturbation extérieure et qui est résistante aux SEU.
- ✓ Un périphérique permettant de récupérer alternativement les valeurs du courant et de la tension d'alimentation du circuit. Les résultats sont accessibles par le bus AMBA grâce à des registres.

11.5 Estimation du taux de SEU par jour

Dossier de référence : */EffetRadiations*

Pour le calcul de probabilité de SEU (*single event upset*), nous avons repris les paramètres donnés dans les slides du cours. La donnée est la suivante :

Spécifications	
• LET = 60 [MeV • cm ² / mg]	
• Flux galactique = 15E-9 [1 / m ² / sr / s]	
• Flux solaire = 20E-6 [1 / m ² / sr / s]	
• Mission: 5 ans, 5 jours d'éruptions solaires	
• Surface pour 1 bit = 200E-9 [cm ² / bit]	
• Nombre de bits:	
– Registres: 4 * 32	
– FSM: 10	
– Compteur: 32	

Nous avons introduit ces données dans le tableau Excel d'exemple et obtenu un taux de SEU de **3.85E-09 SEU/jour**. On a donc une très faible probabilité de SEU.

12 Conception de circuit numériques

12.1 Alimentation FPGA

Dossier de référence : */ConceptionCircuitNumerique*

Voici la liste des tensions d'alimentation et courants nécessaires pour la FPGA Microsemi IGLOO Low Power Flash. Les courants ont été pris pour le modèle AGL015.

Absolute maximum rating				
Symbol	Description	Min	Max	Units
VCC	DC core supply voltage	-0.3	1.65	V
VJTAG	JTAG DC voltage	-0.3	3.75	V
VPUMP	Programming voltage	-0.3	3.75	V
VCPLL	Analog power supply (PLL)	-0.3	1.65	V
VCCI	DC I/O buffer supply voltage	-0.3	3.75	V
VMW	DC I/O buffer supply voltage	-0.3	3.75	V
VI	I/O input voltage	-0.3	3.6	V

AGL015				
Symbol	Description	Min	Max	Units
ICCA	Includes VCC, VPUMP, and VCCPLL currents.	5	14	µA
ICCI	DC I/O buffer supply current	1.7	2.5	µA
IJTAG	JTAG DC current	1.7	2.5	µA

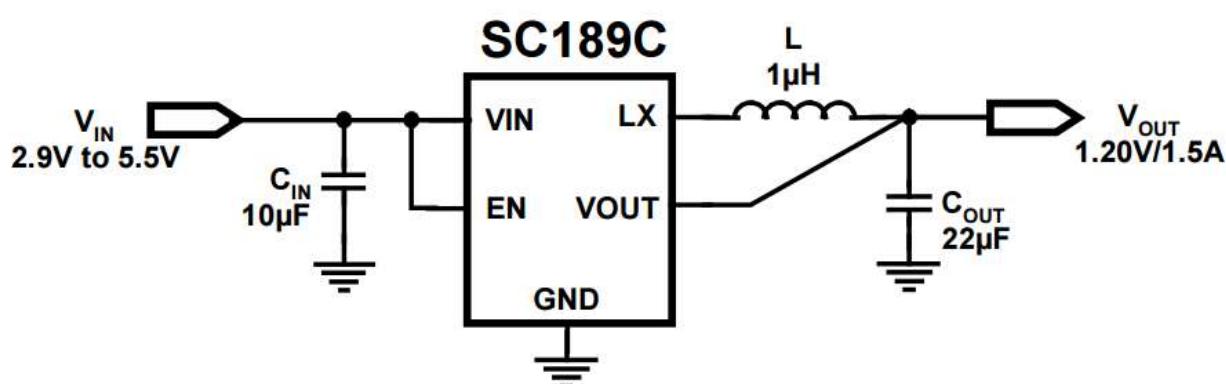
12.1.1 Circuit d'alimentation

À l'aide du tableau ci-dessus, on observe qu'une seule tension d'alimentation peut suffire à alimenter tout le circuit. On pourrait alimenter l'ensemble à 1.2V. Cela donnerait le tableau de courant plus précis suivant :

AGL015			
Symbol	Description	Typ	Units
ICCA	Includes VCC, VPUMP, and VCCPLL currents.	5	µA
ICCI	DC I/O buffer supply current	1.7	µA
IJTAG	JTAG DC current	1.7	µA

IDD=Nbanks x ICCI+ICCA+IJTAG, selon schématique, 4banks => IDD = 13.5 µA

Pour les 1.2V, on peut reprendre le circuit d'alimentation déjà existant dans la schématique de la FPGA du CanSat. Le DC/DC utilisé fournit 1.2V et 1.5A max, ce qui est suffisant pour alimenter notre circuit.



12.1.2 Séquencement

Voici les informations sur la séquence de démarrage trouvées dans le datasheet de la FPGA.

I/O Power-Up and Supply Voltage Thresholds for Power-On Reset (Commercial and Industrial)

Sophisticated power-up management circuitry is designed into every IGLOO device. These circuits ensure easy transition from the powered-off state to the powered-up state of the device. The many different supplies can power up in any sequence with minimized current spikes or surges. In addition, the I/O will be in a known state through the power-up sequence. The basic principle is shown in Figure 2-1 on page 2-4 and Figure 2-2 on page 2-5.

There are five regions to consider during power-up.

IGLOO I/Os are activated only if ALL of the following three conditions are met:

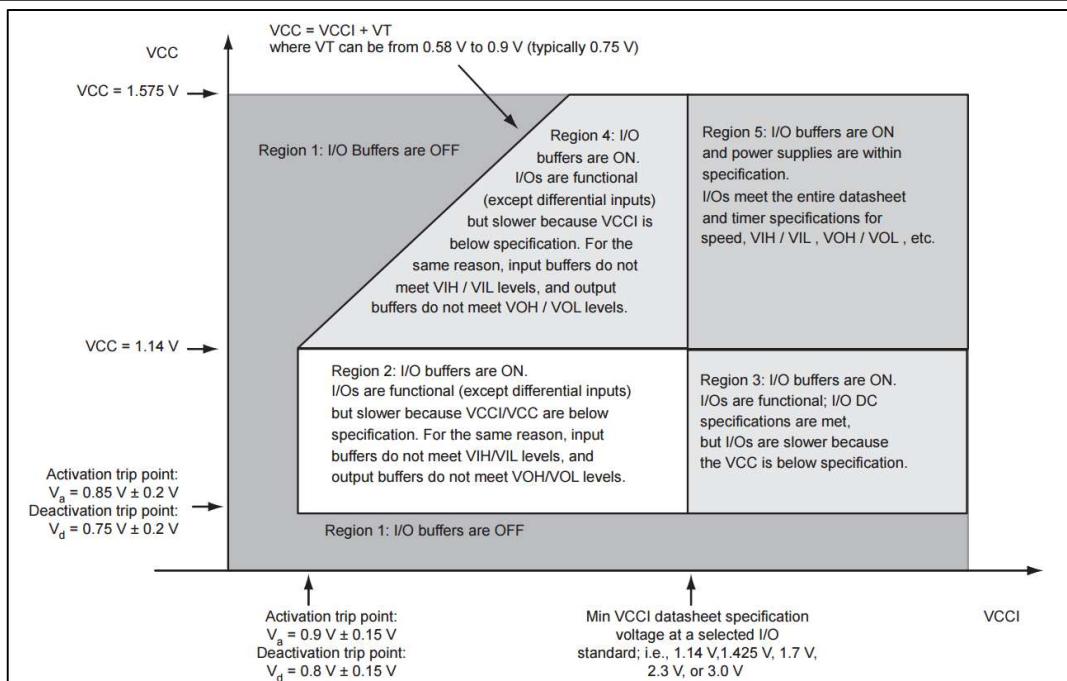
1. VCC and VCCI are above the minimum specified trip points (Figure 2-1 on page 2-4 and Figure 2-2 on page 2-5).
2. VCCI > VCC – 0.75 V (typical)
3. Chip is in the operating mode.

- During programming, I/Os become tristated and weakly pulled up to VCCI.
- JTAG supply, PLL power supplies, and charge pump VPUMP supply have no influence on I/O behavior.

Internal Power-Up Activation Sequence

1. Core
2. Input buffers
3. Output buffers, after 200 ns delay from input buffer activation

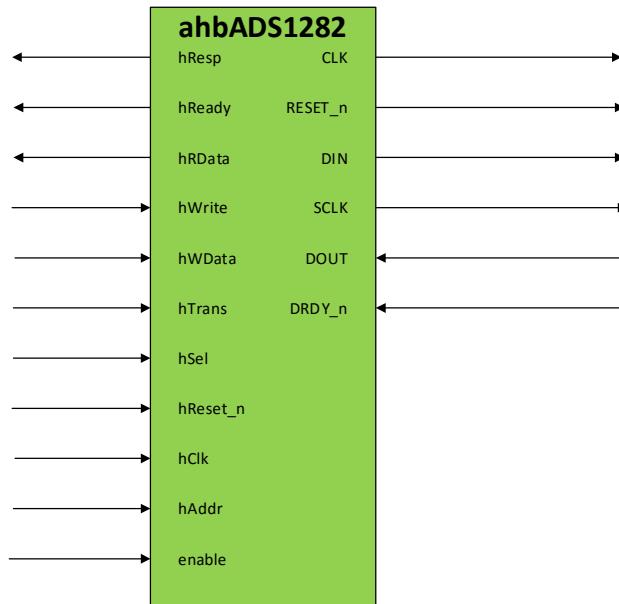
To make sure the transition from input buffers to output buffers is clean, ensure that there is no path longer than 100 ns from input buffer to output buffer in your design.



En résumé, toutes les entrées-sorties sont dans un état connu durant toute la séquence de démarrage du circuit. Lors de la programmation de la FPGA, toutes les entrées-sorties passent dans l'état tri-state.

12.2 Entrées / Sorties du circuit

Voici une représentation du top-level de notre circuit pilotant le convertisseur A/D. Si nous avions eu HDL Designer, on aurait pu visualiser l'intérieur de ce bloc et voir ainsi comment nos différents processus sont reliés entre eux et comment ils sont synchronisés. On ne peut malheureusement pas le montrer.



Dans notre développement, nous avons fait en sorte de synchroniser un maximum de processus sur les signaux de reset et de clock. Toutes nos entrées ont un état stable après un reset. Le clock est le même dans tous nos processus.

13 Vérification de circuits numériques

13.1 Matrice de conformité

Dossier de référence : */VerificationCircuitNumerique* et */VHDL/CanSat*

Le document contient dans sa première feuille la matrice de conformité de notre périphérique de contrôle. Les feuilles suivantes contiennent des captures d'écran ainsi que des remarques sur les différents tests réalisés.

Le code de notre périphérique a dû être adapté, car certains tests ont montré des dysfonctionnements. Le banc de test *ahbAds1282_tester_test.vhd* a également été modifié pour permettre d'effectuer nos différents tests.

14 Tests fonctionnels

Les tests de fonctionnement ont pour but de vérifier le fonctionnement et les performances électriques sous des conditions environnementales représentatives.

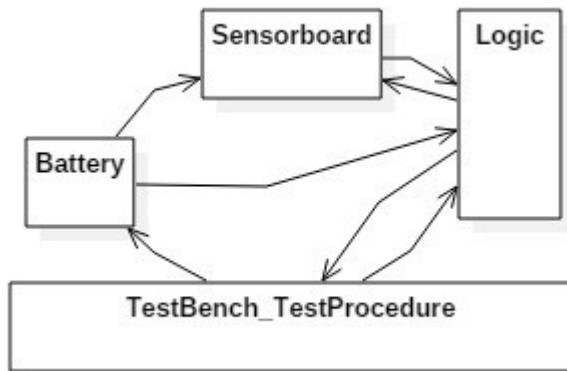
Les tests fonctionnels ont été mis en place en respectant les normes MIL-STD-810 et MIL-STD-202

14.1 Environnement de test

Les tests sont mis en place avec un testeur qui injecte des consignes dans le système pour vérifier le bon fonctionnement de celui-ci. Comme le montre la figure suivante, les tests sont divisés en 3 parties :

- Le testeur, un programme pour lancer les tests, ou un opérateur.
- Une tension d'alimentation réglable

- Un bloc logique pour la communication digital (SPI), pour la configuration et la récupération des données échantillonnées.

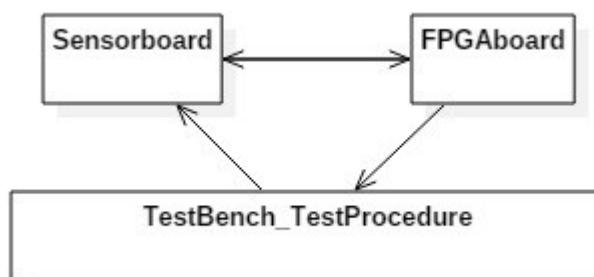


Le testeur permet de mettre en place les tests et de les valider. Il donne des consignes de tension par le biais de la batterie pour tester le signal avec des niveaux de tension différents. Ceci permettra de vérifier le bon fonctionnement du bloc de redondance.

Le testeur spécifie le comportement de la logique pour la configuration de la carte de mesure et les niveaux de consommation pour des tests de puissance de la carte.

Dans ce cas de figure seules la mesure de courant/tension et la conversion analogique digitale sont contrôlées.

Pour le FTDI et la communication avec la FPGA (SPI et programmation), un deuxième teste sera effectué.



Avec ce test, la communication avec la FPGA (SPI, FTDI et alimentation), pour être vérifiée. L'alimentation sera fournie pour le test via une batterie.

Ces tests peuvent être faits avec différentes perturbations environnementales :

- EMC
- vibration
- thermiques

14.2 Plan de test

Le plan de test pour le CanSat est le suivant :

Tous les types d'environnement doivent être testés selon les normes.

1. Pour l'EMC : avec des niveaux de tensions différentes sur les surfaces métalliques que fer, avec contact et sans, ...
2. Pour les tests de vibration, sur plaque vibrante avec différente vitesse de tours minute
3. Et pour les tests en température, dans un four pour atteindre les 100°C et faire chuter la température rapidement pour atteindre les -40°C.

12 Avril 2016

Environnement	Équipement	Spécification	Montage	Résultats	Critère d'acceptance
1,2,3	USB		Alimentation par l'USB		
1,2,3	Batterie	Tension d'alimentation stable (5V, 3.3V)	Alimentation de la carte avec la batterie, et mesure au niveau des points de test		
	Batterie, CanSat complet	Alimentation des autres cartes	Contrôle de la tension d'alimentation sur les autres cartes.		
	Batterie, FPGA board	Communication FPGA, FTI	Branchemet du l'USB et vérification de la communication		
		Mesure shunt	Alimentation de la carte avec la batterie, Mesure de la tension et du courant après l'amplificateur		
		Conversion courant-tension	Mesure du bon fonctionnement de l'ampli op		
		Communication FPGA (SPI)	Vérification des lignes séries entre la FPGA et l'ADC		
		Configuration de l'ADC	Vérification de la bonne configuration de l'ADC		

D'après les critères d'acceptance il devient possible de faire une déclaration de conformité.

14.3 Procédure de test

Les explications et la réalisation ont été les suivants :

Spécification testée	Montage du test	Résultats obtenus
Bloc alimentation	Connexion de la batterie (et USB) et vérification des tensions d'alimentations 5V, 3.3V aux différents points de test	
Bloc amplification (courant/tension)	Contrôle de la bonne conversion tension courant et de la tension à l'entrée du convertisseur analogique digital	
Bloc conversion AD	Contrôle de la tension de référence, du filtre FIR sur les pins CAPP et CAN, paramétrage des registres du convertisseur et contrôle de l'échantillonnage sur le protocole SPI	
Bloc FPGA (FTDI)	Vérification de la communication série et de la programmation de la FPGA.	
Mise en commun des parties et vérifications complète	Branchemet complet du CanSat et mise en route de toutes les parties	

Cette procédure de test est à améliorer. Si une partie citée ci-dessus ne fonctionne pas correctement, il est nécessaire de faire des étapes supplémentaires pour trouver l'origine de la panne.

Remarques : Ces tests n'ont malheureusement pas pu être réalisés physiquement car l'électronique n'a pas été réalisé, le PCB n'est pas imprimé.

15 Conclusion

Après toutes ces analyses on remarque qu'il est très contrôlé d'envoyer un système en orbite et qui respecte les normes militaires et spéciales. Pour ce projet, nous avons dû choisir un instrument de mesure, à partir de celui-ci faire une schématique, faire des analyses de budgets, chaleurs, risques, Ensuite vient la partie PCB, avec routage des pistes de façon à diminuer les perturbations et de même pour la partie programmation en codant des états pour anticiper des changements involontaire de bits.

D'après ces analyses, nous avons contrôlé et évalué notre système pour un fonctionnement spatial. Ces analyses nous ont permis de bien comprendre notre système et de détecter quelques mauvais choix ou mauvais dimensionnement.

Notre système est pour le moment toujours cloué au sol, il faut tirer le PCB monter les composants, programmer la FPGA et remplir, le fichier testes. Le système telle quelle est fonctionnelle, mais pourrait être amélioré.

Dans une orientation future, il serait envisageable de reprendre toutes les analyses et de modifier la schématique, le routage et la programmation en tenant compte des remarques inscrites en dessous de celles-ci, ce qui permettrait d'améliorer le comportement en vol du système et de garantir une meilleure stabilité.