

Melhoria do ambiente computacional EDA para projetos de CIs com ênfase em aplicações espaciais.

Paula Cristiane Secheusk (CTI) paula.secheusk@cti.gov.br

Resumo

Este trabalho é seqüência e complementar ao apresentado em 2019 e visa apresentar dentre os nós tecnológicos adotados pelo CTI, em seu ambiente de projetos, a metodologia recomendada pelos fabricantes e a customização necessária para adequar o ambiente a cada necessidade singular. Os fluxos de projeto têm uma forte interoperabilidade entre as ferramentas EDA (Electronic Design Automation): Cadence - projeto e simulação, Mentor Graphics - verificação física e extração de parasitas, MunEDA - otimização de resultados; e o Subversion (SVN) e o TortoiseSVN (TSVN) no controle de versão. O uso das ferramentas de projeto profissionais é imprescindível para atingir os objetivos de realizar projetos corretos por construção (design correct by construction).

Como resultados foram customizados: dois tipos de fluxo de projeto para CIs, dois tipos de formulários relacionados a contas de usuários, quatro checklists, quatorze scripts em linguagem Shell Script para a customização e gerenciamento dos ambientes Linux x Projetos x Controle de versão, nove scripts em linguagem SKILL para a adição de funcionalidades extras do software de controle de versão Subversion (SVN) às ferramentas Cadence (View Menu), um cliente Subversion (TSVN) para Microsoft Windows (controle de versão), e três rotinas de backup.

Um ambiente estável orientado pelos fabricantes de tecnologia é primordial. A melhoria do ambiente computacional EDA para projeto de CIs (Circuito Integrado) permite o desenvolvimento de circuitos na metodologia recomendada pelas foundries. No âmbito do Projeto CITAR (Circuitos Integrados Tolerantes à Radiação) foram desenvolvidos vários circuitos, os quais foram fabricados e estão sendo testados conforme as normas internacionais, obtendo-se excelentes resultados.

Palavras-chave: Metodologia de Projeto; ASICs; EDA; Radiação ionizante e Circuitos Integrados.

1. Introdução

Os constantes avanços tecnológicos na indústria de semicondutores têm permitido minimizar as dimensões das células elementares dos dispositivos eletrônicos, aumentando sua densidade e conseqüente funcionalidade. Estes avanços determinam constantes atualizações nos métodos e procedimentos desenvolvidos para modelagem, projeto e fabricação dos circuitos integrados. A Figura 1 adaptado de [SPINOGRAPH], ilustra a evolução dos nós tecnológicos da tecnologia CMOS, onde são ressaltados em cor verde os nós usados no Núcleo de Concepção de Sistemas de Hardware (NCSH) do CTI para o desenvolvimento dos projetos.

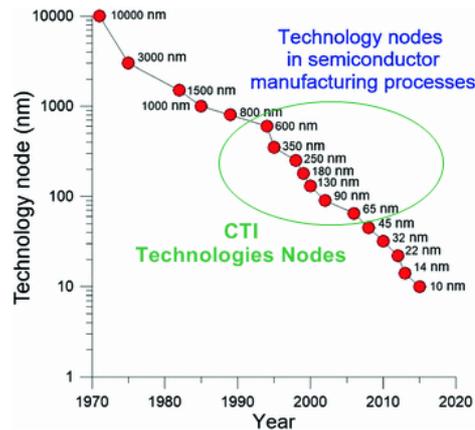


Figura 1 – Nós Tecnológicos do NCSH-CTI

Portanto, atualizar as facilidades do ambiente de projetos do CTI para realizar desenvolvimento em sistemas e CIs, é primordial para preservar a relevância da instituição como referência em ambiente de projetos de CIs no âmbito nacional, bem como, tornar-se um centro de referência mundial em treinamento e divulgação no estado da arte dos desenvolvedores de ferramentas EDA.

2. Objetivos e Metodologia

Para se iniciar um projeto de CI é necessária a definição do nó tecnológico, e conseqüentemente, o *Process Design Kit* (PDK) a ser usado de um determinado fabricante. No PDK são especificados os tipos de ferramentas EDA qualificadas (Cadence, Mentor Graphics, MunEDA, etc) a serem usadas e suas versões, além da definição dos Sistemas Operacionais [CADENCE S.O.] suportados, Figura 2[FIGHW].

De posse destas informações, faz-se a verificação dos Sistemas Operacionais suportados pelo hardware disponível no ambiente de produção (estações de trabalho e servidores), haja visto que esse hardware pode ser um grande limitante em desempenho, capacidade de processamento e armazenamento (CPU, RAM, e HDD).



Figura 2 – Fluxograma das necessidades operacionais para projeto de CIs.

Deste modo, o ambiente de projetos deve proporcionar uma perfeita sinergia entre as variantes, sejam elas de: hardware, sistemas operacionais, ferramentas EDA e PDKs. E, devido a forte interoperabilidade entre as ferramentas de projetos de CIs, elas podem ser integradas para garantir o fluxo de desenvolvimento: Cadence – projeto e simulação [CADENCE1-3], Mentor Graphics – verificação física e extração de parâmetros parasitários [CALIBRE1-2, PYXIS], MunEDA – otimização de resultados; e, ferramenta de controle de versão [SVN1-2, TORTOISESVN].

Cada etapa do fluxo de projeto demanda ferramentas distintas (Figura 3), e é relevante destacar que através do desenvolvimento de metodologias e customizações para o aprimoramento na forma de trabalho dos projetistas, organização da base de dados e da documentação, controle de versão, e rotinas de *backup* são atividades imprescindíveis à manutenção de um ambiente de projeto, garantindo a integridade dos dados obtidos ao longo de todo o processo de desenvolvimento em suas mais diversas etapas.

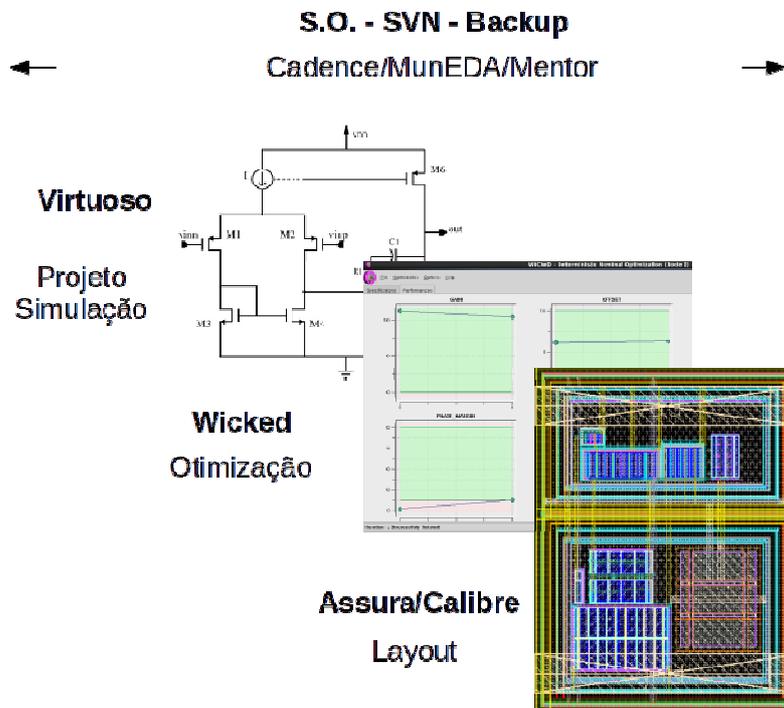


Figura 3 – Fluxo de projeto de CIs.

3. Resultados

A estrutura e metodologias propostas neste trabalho têm permitido o desenvolvimento das atividades em execução de projetos de CIs no NCSH, nas melhores práticas recomendadas. Temos reiterado ao longo da pesquisa deste trabalho, que é imprescindível a atualização constante dos procedimentos recomendados nos projetos de CIs, levando-se em consideração as necessidades do mercado em geral e no nosso caso, em aplicações espaciais e similares. Assim sendo, seguem alguns resultados:

3.1. Interação PDKs x Ferramentas EDA x Sistemas Operacionais

A escolha do Sistema Operacional do trabalho anterior continua válida, bem como a sua integração com os diferentes PDKs/*Foundries* e as ferramentas EDA, ou seja, têm instalado no NCSH: uma versão de Sistema Operacional (RHEL 6.5 x86_64bits), mantendo-se a versão RHEL 5.1 x86_64bits para projetos antigos; 10 PDKs, sendo uma versão TSMC-65nm e três versões TSMC-180nm [TSMC], UMC-180nm [UMC], duas versões XFAB-XT018 [XT018], duas versões XFAB-XT06 [XT06], uma versão XFAB-XC06 [XC06]; e inúmeras ferramentas EDA com suas respectivas atualizações.

3.2. Desenvolvimento de Métodos e Procedimentos para melhoria do ambiente EDA para projeto de circuitos espaciais

Para atender as novas demandas dos projetos do NCSH e dar continuidade as melhorias do ambiente EDA, foram desenvolvidas: dois tipos de formulários para gerenciamento de usuários: Abertura de Conta de Usuário e Desligamento de Usuário; quatro tipos de *checklists* contendo o

passo-a-passo para: Abertura de Conta de Novo Usuário, Remoção de Conta de Usuário, Abertura de Novos *Wafers* e Inclusão de Novos Usuários à *Wafers*; dois tipos de fluxos de projeto para CIs: Área de Desenvolvimento e Área para Propriedade Intelectual (IP): uma estrutura do repositório do projeto para o SVN e dos diretórios para a configuração do projeto, uma estrutura de diretórios de trabalho para circuitos Analógicos/RF e Digitais, padronização da nomenclatura de blocos para as bibliotecas Cadence/Mentor, dois documentos descritivos: Utilizando um IP em Projetos, e, Transformando um Bloco de Projeto em IP.

A Figura 4, por exemplo, ilustra a estrutura de diretórios para o Fluxo de Projeto de IP para Circuitos Digitais.

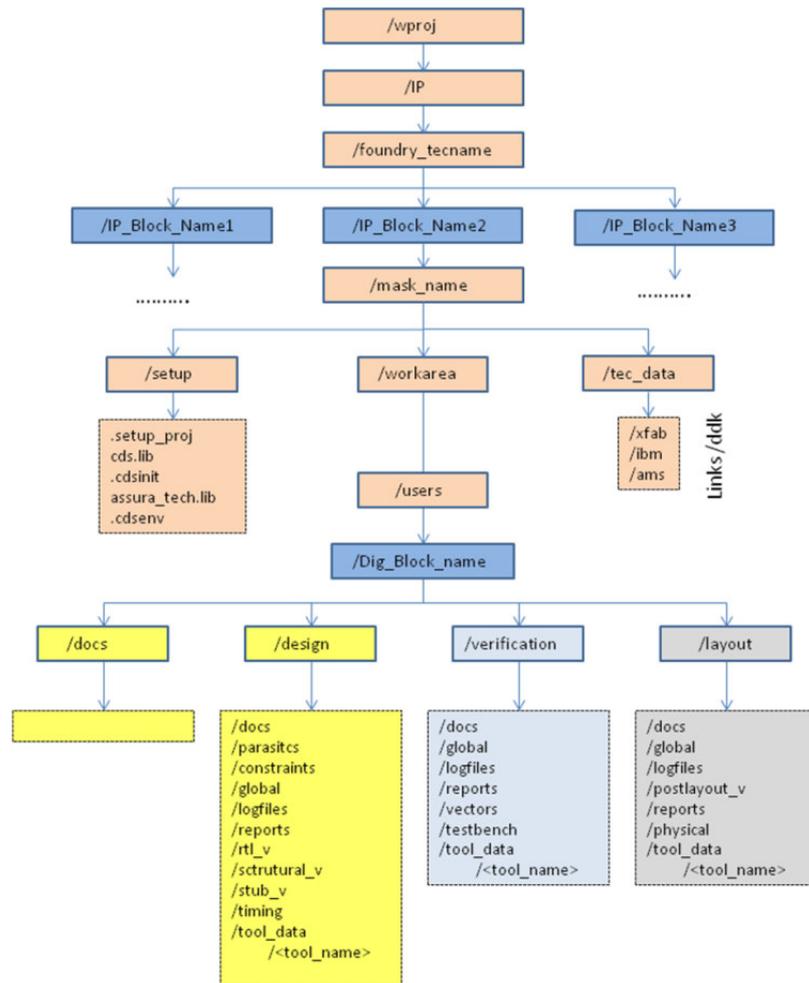


Figura 4 – Estrutura de diretórios para o Fluxo de Projeto de IP para Circuitos Digitais

3.3. Interação Controle de Versão x Ferramenta EDA x Backup

A escolha do uso da ferramenta de controle de versão - *Subversion* (SVN, para linux), software livre da *CollabNet VersionOne* continua válida no NCSH, garantindo a integridade dos dados obtidos ao longo de todo o processo de desenvolvimento de um projeto de CI.

No trabalho anterior, foi implementado controle de versão dos blocos (*cells*) em desenvolvimento, e para atender as novas demandas dos projetos que requerem também o controle de versão separado de cada um dos sub componentes internos (*views*), fez-se necessário o estudo e desenvolvimento de novos *scripts* em linguagem *SKILL*, para a integração da ferramenta *Subversion* à ferramenta Cadence, resultando na criação de dez novos *scripts* que adicionam funcionalidades extras do SVN às ferramentas Cadence (*View Menu*). A Figura 5 ilustra esta integração.

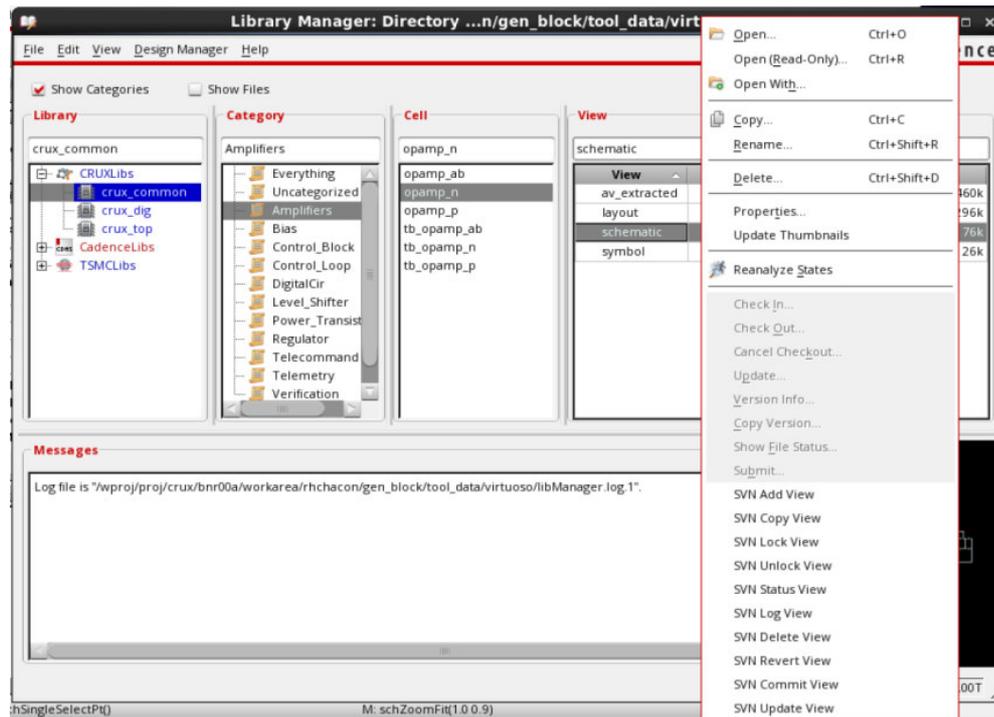


Figura 5 – Integração da ferramenta Subversion à ferramenta Cadence (ViewMenu)

E, com intuito de facilitar algumas atividades do *SVN Admin*, fez-se necessário o estudo e desenvolvimento de quatorze *scripts* em linguagem *Shell Script*, para a customização e gerenciamento das atividades relacionadas à administração do SVN x Projetos do NCSH. A Figura 6 ilustra os scripts e suas funcionalidades.

	Scripts SVN-Admin:	Funcionalidades:
1	<i>svn_adduser</i>	Adiciona um USER as bases de autenticação SVN, Joomla, Openfire, Redmine, NCSH.
2	<i>svn_changeuser</i>	Altera a senha de um USER as bases de autenticação SVN, Joomla, Openfire, Redmine, NCSH.
3	<i>svn_checkexists</i>	Verifica a existência de um USER nas bases de autenticação SVN, Joomla, Openfire, Redmine, NCSH.
4	<i>svn_deluser</i>	Remove um USER das bases de autenticação SVN, Joomla, Openfire, Redmine, NCSH.
5	<i>svn_lslocks</i>	Lista todos os arquivos travados (lock) de um repositório. Deve ser executado no Servidor SVN.
6	<i>svn_randompass</i>	Gerador de senhas aleatórias.
7	<i>svn_rmlocks</i>	Lista e remove todos os arquivos travados de determinado repositório/usuário.
8	<i>svn_searchignores</i>	Busca no repositório arquivos que não deveriam estar sob controle de versão.
9	<i>svn_searchuser</i>	Busca na base de repositórios, nos arquivos authz, ocorrências de um nome de USER.
10	<i>svn_showmrepos</i>	Lista todos as URLs de todos os repositórios.
11	<i>svn_verify_repo</i>	Verifica se existem inconsistênciasna base de dados de um repositório.
12	<i>urlaf</i>	Exibe a URL de determinado PATH em uma área de trabalho SVN.
13	<i>confbck</i>	Realiza backup dos arquivos de configuração do SVN. Deve ser executado no Servidor SVN.
14	<i>statgen</i>	Gera estatísticas e gráficos HTML de todos os repositórios Subversion a partir da 2ª revisão.

Figura 6 – Listagem dos Scripts para à administração do SVN x Projetos do NCSH

Apesar da maioria das ferramentas EDA usadas no fluxo de projeto de CIs serem para sistema operacional Linux, há ferramentas que precisam ser instaladas no sistema operacional Microsoft Windows. Portanto, após estudos e o levantamento das necessidades do NCSH, optou-se também por fazer uso do software *TortoiseSVN* que é um cliente *Apache Subversion* para Microsoft Windows cujo acesso principal é via Windows Explorer. A ferramenta foi instalada e configurada em todas as máquinas virtuais Windows do NCSH, o que ocasionou o desenvolvimento de um treinamento para o time de projetistas do Núcleo. A Figura 7 ilustra a arquitetura básica do repositório *Subversion* com a integração da ferramenta *TortoiseSVN*.

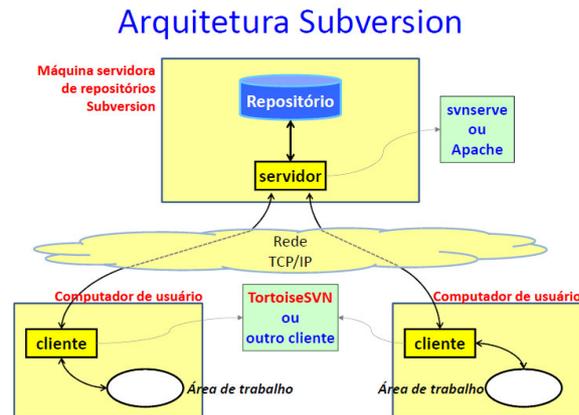


Figura 7 – Arquitetura Subversion com integração do cliente Apache Subversion - TortoiseSVN

Dada à complexidade da estrutura de projetos de CIs num ambiente computacional EDA, fez-se necessário também o estudo e desenvolvimento de novos *scripts* em linguagem *Shell Script*, para a realização de rotinas simples de *backup* de uso exclusivo do NCSH, gerando a criação de três novos *scripts*: *backup* do *homedir* dos usuários - /export/users/, *backup* dos projetos antigos do Núcleo - /proj/ e *backup* das *workareas* de projetos - /wproj/, os quais são todos gerenciados pelo arquivo *crontab* localizado no servidor *seberi.cti.gov.br*.

3.4. Ambiente de projeto confiável proporciona publicação de resultados

Adicionalmente aos resultados anteriormente descritos, ressalta-se que a melhoria do ambiente computacional EDA, tem permitido desenvolver circuitos integrados usando as estruturas propostas neste trabalho. Exemplo disto é o projeto SpaceWire desenvolvido no Núcleo seguindo o fluxo digital, o qual atingiu as especificações requeridas. Alguns resultados desse trabalho são apresentado no item 1 desta seção. Além disso, a estrutura para circuitos analógicos permitiu o desenvolvimento de um sistema de proteção de cargas úteis para aplicações espaciais, por meio da limitação de corrente. A descrição do sistema, implementação e resultados de teste são apresentados no item 2 como segue:

1. DEUCHER A ; SANTOS, Â. A. ; LAZARI, D. B. ; FINCO, S. . SpaceWire Receiver: Synchronization for Different Clock Domains. Proceedings of the 4th Brazilian Technology Symposium (BTSym'18) Emerging Trends and Challenges in Technology. 1ed.Cham: Springer Nature Switzerland AG, 2019, v. 140, p. 35-44.
2. Chacón, R. H. G., Dias, A. V., Santos, Â. A., Secheusk, P. C., Manea, S., Diniz, J. A., Finco, S. (2020). A Latching Current Limiter with Telemetries for Space Application. Manuscrito em preparação a ser submetido a revista indexada.

4. Conclusão

A customização do fluxo de projeto em cada uma das suas etapas, demanda a integração de diferentes ferramentas. Além disso, o tipo de tecnologia também define o uso de aplicações específicas, como é o caso da ferramenta Calibre usado para a verificação física por algumas *foundries*. Os *scripts* desenvolvidos permitiram criar uma estrutura mais dinâmica para os projetistas, auxiliando na identificação dos problemas e agilizando sua solução.

Neste contexto, a pesquisa contínua e o desenvolvimento do trabalho aqui apresentado têm garantido a confiabilidade das bases de dados geradas nos projetos desenvolvidos, enviados para fabricação e testados no Núcleo. Tem permitido também que novos projetos com diferentes tecnologias (PDKs) possam ser executados. Exemplo disto são os projetos implementados no âmbito do projeto CITAR na tecnologia *Silicon On Insulator* (SOI) de 600nm da XFAB prontos para serem fabricados, e na tecnologia *Bipolar-CMOS-DMOS* (BCD) 180nm da TSMC atualmente em fase de desenvolvimento, aplicando técnicas de mitigação de efeitos em estruturas nos projetos

para aumentar a robustez à radiação ionizante dos CIs. Este ambiente estável e atualizado tem permitido aos projetistas, a confiança em seus resultados para publicação destes junto a comunidade acadêmica e científica.

5. Referências

- AHSAN JAFRI, E. PAUL JEBASINGH, SHANTAGOUDAR GOUDAR** *EDA cost optimization through design process analytics*. 2016 Sixth International Symposium on Embedded Computing and System Design (ISED), Conference Paper, p. 340-344, 2016.
- BOWU YAN, XICAI CHENG, FEI YANG, LI YAO** *Research on EDA technology and its related issues*. 2010 International Conference On Computer Design and Applications, Vol. 4, p. 26-29, 2010.
- CADENCE1** Manual: Virtuoso AMS Environment User Guide, V. 5.3, April 2004: Cadence_AMS_Environment_User_Guide.pdf
- CADENCE2** Manual: Cadence Library Manager User Guide, V. 4.4.6, June 2000: Cadence_Library_Manager_User_Guide.pdf
- CADENCE3** Manual: Cadence Application Infrastructure User Guide, V. 3.4, November 2005: Cadence_Application_Infrastructure_User_Guide.pdf
- CADENCE S.O.** Computing Platforms - Supported Platform Matrix for Cadence Applications, July 2019: platform.pdf
- CALIBRE1** <https://support.sw.siemens.com/en-US/knowledge-base/MJ584482>
- CALIBRE2** http://calibre.mentorcloudservices.com/docs/Calibre_OS_Roadmap.htm
- FIGHW** Apresentação “Computador”, Adão de Melo Neto, arquivo: MEMORIA_2017.pdf
- PYXIS** https://support.sw.siemens.com/en-US/product/852852102/downloads?downloadId=DL201404084en_US
- SKILL1** Manual: Cadence SKILL Language Programming, V. 5.0, July 26, 2002: Skill_Language_Programming.pdf
- SKILL2** Manual: SKILL Language Reference, V. 06.30, November 2008: Skill_Language_Reference.pdf
- SKILL3** Manual: SKILL Language User Guide, V. 06.30, September 2005: Skill_Language_User_Guide.pdf
- SPACEWIRE1** LAZARI, D. B. ; DEUCHER A ; SANTOS, Â. A. ; FINCO, S. ; HORN, A. ; BEER, M. ; OHLEN, V. Debug and Verification of SpaceWire links Test and Verification Session, Short Paper, 2018, In: International SpaceWire Conference 2018, 2018, Los Angeles.
- SPACEWIRE2** A. DEUCHER, A. A.SANTOS, J.L. EMERI JR., D.B. LAZARI, A.C. C. TELLES AND S. FINCO "Radiation Hardened by Design library and Triple Module Redundancy with Error Correction Codes for development Space Wire ASIC". SEMINATEC 2019.
- SPINOGRAPH** <http://www.spinograph.org/blog/why-nanoelectronics-better-microelectronics>
- SVN1** Manual: Version Control with Subversion for Subversion 1.6 (Compiled from r3765), Ben Collins-Sussman, Brian W. Fitzpatrick, C. Michael Pilato: SVN_Book.pdf
- SVN2** Manual: Subversion v1.6 – Quick Reference Card: Subversion_Quick_Reference_Card.pdf
- TORTOISESVN** <https://tortoisesvn.net/>
- TSMC** TSMC PDK Checklist – 0.18um MIXED SIGNAL GENERAL PURPOSE II 1P6M/1P5M SALICIDE 1.8V/3.3V PDK, V. 1.3A, 01/17/2017: tsmc18mm_1.8v_3.3v_FSA_checklist_1.3a.pdf
- UMC** UM180FDKMFC000000A Foundry Design Kit of 0.18 um Mixed-Mode and RFCMOS 1.8V/3.3V 1P6M Metal Metal Capacitor Process, V. A02_P.B, 2014/04/16: G-9FD-MIXED_MODE_RFCMOS18-1.8V_3.3V-1P6M-MMC_UM180FDKMFC000000A-FDK-Ver.A02_PB.pdf
- XC06** GSA MIXED-SIGNAL/RF, PDK CHECKLIST, V. 3.0, XC06 – 0.6um Modular CMOS (XC06), 01/15/2014: xc06-GSA_PDK_Checklist-Mentor_Pyxis-v4_1_3.pdf
- XT06** GSA MIXED-SIGNAL/RF, PDK CHECKLIST, V. 3.0, XT06(M2), XT06M3, 04/15/2011: xt06-GSA_PDK_Checklist-Cadence_IC61-v2_1.1.pdf
- XT018** GSA MIXED-SIGNAL/RF, PDK CHECKLIST, V. 3.0, XT018 – 0.18um HV SOI CMOS, 09/23/2016: xt018-cds-PDK-GSA_Checklist-v5.0.1.pdf