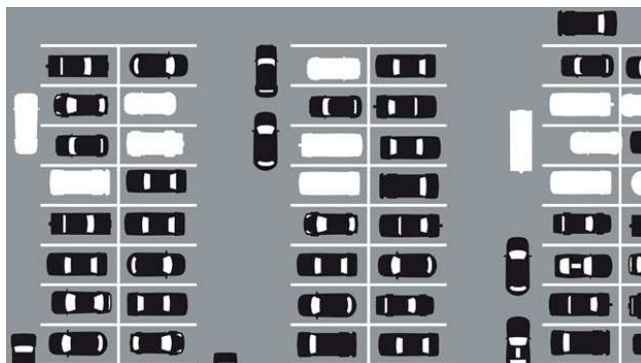




Sciences et technologies de l'Industrie et du développement durable

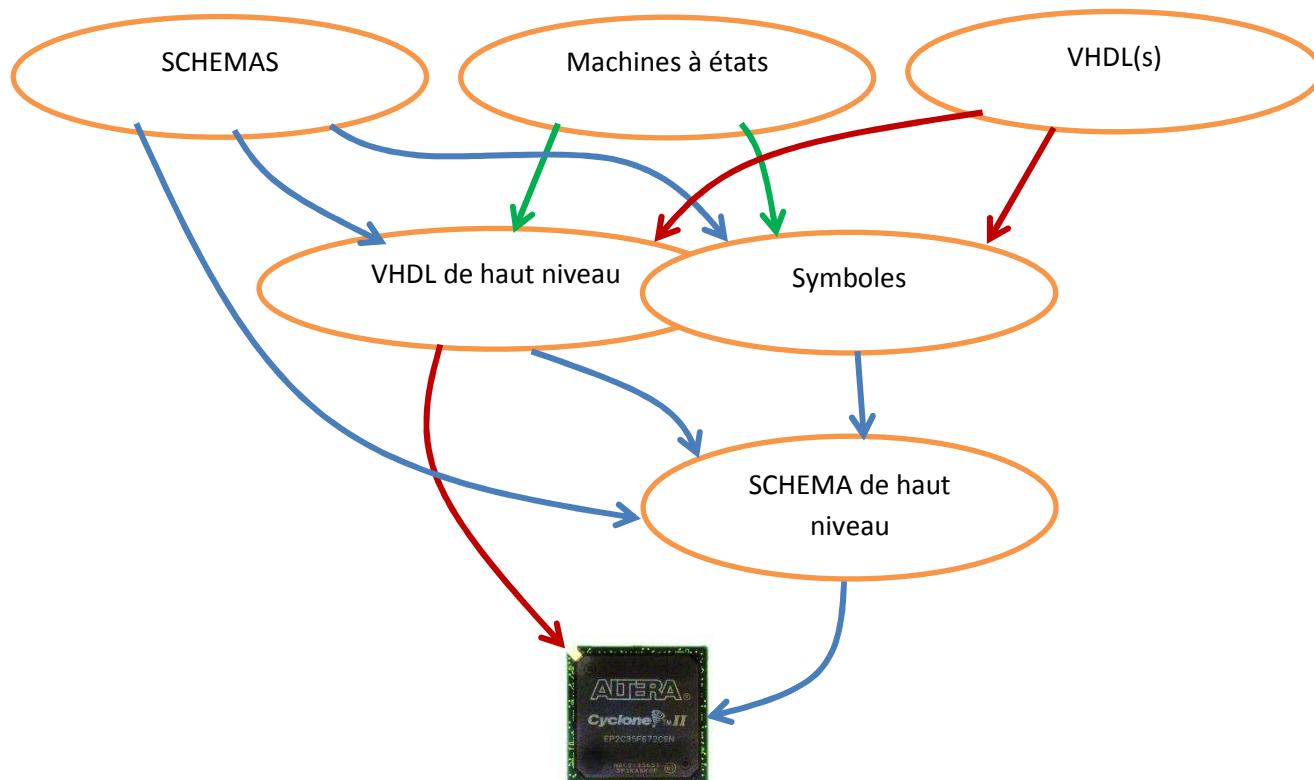
Le projet « gestion de parking »

Le parcours est architecturé autour de l'étude d'un gestionnaire simple de parking. Il permet une première approche des différents procédés de développement d'un projet dans un FPGA.



- Par schéma fonctionnel/logique
- Par machine à états (MAE)
- En langage VHDL (le langage Verilog n'est pas abordé)

Les diverses fonctions d'un projet peuvent être développées par une équipe de techniciens/ingénieurs dans n'importe lequel des procédés ci-dessus. Les représentations par schéma ou par MAE étant transformées en symboles électriques et/ou en langage VHDL. L'outil de développement assemblera les descriptions VHDL et s'efforcera de les router dans le composant cible. Quartus donne ainsi beaucoup de souplesse pour le développement des applications.



L'objectif est de réaliser complètement le projet « gestion de parking » proposé ci-après, sa simulation ainsi que son implantation sur une carte DE2 (voir descriptif des équipements dans Carte_parcours_SIN_FPGA.doc.) et procédure d'installation (Installer QUARTUS et MODELSIM.docx)

Pour cela l'apprenant peut s'aider des trois travaux pratiques proposés ci-dessous qui offrent un apprentissage progressif des trois modes d'entrée d'une description (design) dans QUARTUS II d'ALTERA avec à chaque fois une série d'exercices.

SIN FPGA TP1

- Prise en main de l'outil, création d'un projet
- Description des différentes phases de développement et de synthèse
- Apprentissage de la description par schéma
- Prise en main du simulateur ModelSim
- Implantation dans le composant (sur carte DE2)

Ce TP est indispensable aux nouveaux utilisateurs de QUARTUS. Il présente les concepts de développement d'un FPGA, de simulation comportementale et matérielle.

Il comporte l'apprentissage de base des outils de développement d'ALTERA.

SIN FPGA TP2

- Apprentissage de la description par machine à états.

Ce TP comporte de nombreux exercices et met en place les différences fondamentales d'utilisation et de mise en œuvre des simulations comportementales et matérielles dans ModelSim.

SIN FPGA TP3

- Apprentissage de la description en langage VHDL

Ce TP est une approche de la description en langage VHDL, le lecteur se réfèrera aux ouvrages conseillés pour l'apprentissage complet du langage.

Il n'est cependant pas nécessaire d'avoir des connaissances approfondies en langage VHDL pour réaliser un projet FPGA, celui-ci pouvant être décrit en schéma et MAE..

La simulation comportementale étant réalisée à partir de fichiers VHDL ou Verilog, il est indispensable d'avoir les connaissances de lecture et d'analyse des fichiers VHDL générés par QUARTUS.

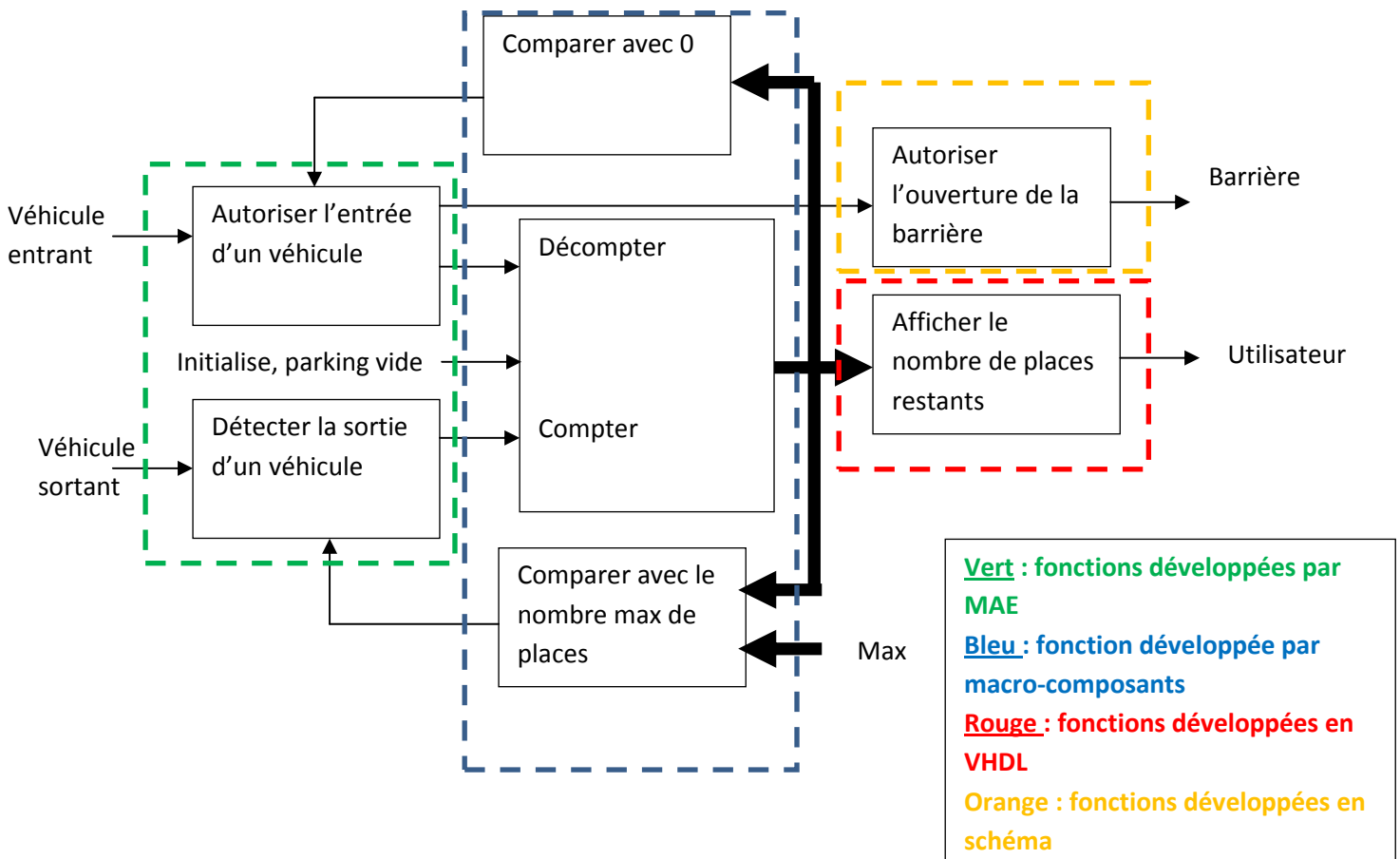
Dans de nombreux cas la description VHDL s'avèrera beaucoup plus efficace que la description par schéma.

Une vérification du fichier VHDL généré par le compilateur de machine à états s'avère systématiquement obligatoire.

Pour apprendre le VHDL, consulter l'ouvrage gratuit de J.Weber et S.Moutault :

<http://books.google.fr/books?id=AKolOwjcnUC>

Schéma fonctionnel du gestionnaire de parking :



Les véhicules sont détectés en entrée et en sortie (*radars ou boucles magnétiques*), un niveau logique haut indique la présence d'un véhicule sur le capteur.

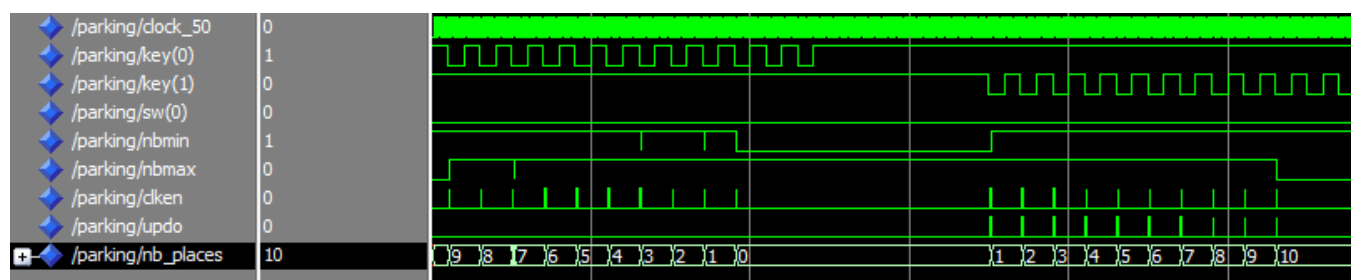
En entrée l'autorisation de passage a lieu si un véhicule est présent et s'il reste au moins une place dans le parking. Un niveau logique 1 est alors envoyé à la commande de la barrière durant 1 seconde, le nombre de places disponibles est décompté.

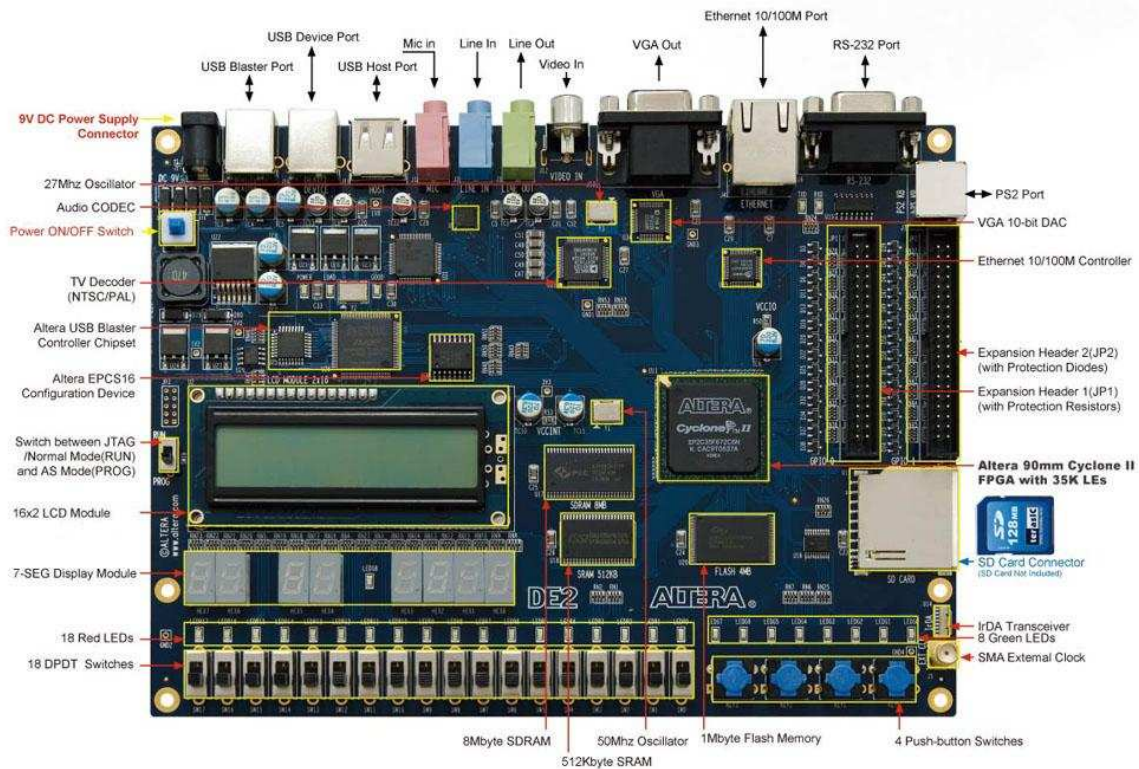
En sortie, lors de la détection du passage d'un véhicule le nombre de places dans le parking est incrémenté sans dépasser le nombre maximum de places du parking (Max).

Une initialisation est possible lorsque le parking est vide, le nombre de places disponibles est alors égal au maximum.

Le nombre de places disponibles (999 ou moins) est affiché en permanence sur trois afficheurs 7 segments à LEDs.

- Les essais se feront sur une carte DE2.
- Les touches KEY[0] et KEY[1] représenteront le passage des véhicules en entrée et en sortie du parking.
- SW[0] permettra la réinitialisation du comptage au maximum.





Altera Cyclone II 2C35 FPGA with 35000 LEs

- Altera Serial Configuration devices (EPCS16) for Cyclone II 2C35
- USB Blaster built in on board for programming and user API controlling
- JTAG Mode and AS Mode are supported
- 8Mbyte (1M x 4 x 16) SDRAM
- 512K byte(256K X16) SRAM
- 4Mbyte Flash Memory (upgradeable to 4Mbyte)
- SD Card Socket
- 4 Push-button switches
- 18 DPDT switches
- 9 Green User LEDs
- 18 Red User LEDs
- 16 x 2 LCD Module
- 50MHz Oscillator and 27MHz Oscillator for external clock sources
- 24-bit CD-Quality Audio CODEC with line-in, line-out, and microphone-in jacks
- VGA DAC (10-bit high-speed triple DACs) with VGA out connector
- TV Decoder (NTSC/PAL) and TV in connector
- 10/100 Ethernet Controller with socket.
- USB Host/Slave Controller with USB type A and type B connectors.
- RS-232 Transceiver and 9-pin connector
- PS/2 mouse/keyboard connector
- IrDA transceiver
- Two 40-pin Expansion Headers with diode protection

