

Apport des circuits logiques dans la formation informatique

Commentaires aux chapitres 1-2-3 www.didel.com/LogiChap1.pdf etc

Les logidules EPFL et les kits distribués à fin 2019 ont mis en avant une vision trop purement technologique des circuits logiques.

Dans la documentation 2020, l'approche n'est plus "câblons et comprenons" mais "intéressons-nous et pour assimiler, câblons, simulons et programmons".

Il faut donc s'efforcer de mettre en évidence des concepts et notions importantes et de faire réfléchir, inventer les étudiants. Pour certains points des compléments de 2-4 pages vont en profondeur et des projets de maturité sont suggérés.

Un effort a été fait pour les figures mais la rédaction est un premier jet sans talent pédagogique. Si ce document présente de l'intérêt, il est à revoir complètement par des enseignants qui sauront s'adapter au niveau de leurs étudiants.

Notions vues dans le chapitre 1 Nombres et logique combinatoire

Rang et poids dans un système de numération - 1.1

Dans quelle base sont exprimés les nombres de la base x ? Convertir de x à y est alors évident. Les étudiants savent que tout se calcule en binaire dans l'ordinateur, avec conversion en décimal quand on tape un nombre ou affiche une valeur?

Diviser pour régner - 1.2

L'addition se décompose en modules qui ont la même fonctionnalité, mais il y a toujours des "conditions de bord" à gérer différemment

Signaux et variables informatiques - 1.3

Les notations C ou Python aident-elles pour concrétiser les variables booléennes (le report) et les variables. Vrai-faux, 0-1, 0-36, comment faire sentir les différences?

Commutateurs et courant électrique - 1-5

Analyser un circuit dans lequel le courant passe d'une tension positive à négative a été vu. Et probablement oublié.

Notion de fiabilité des transferts de signaux dans un ordinateur - 1.6

Il faut respecter des marges de sécurité, en tension et en temps (worst case design repris en 2.6).

Portes logiques - 1.7 à 1.11

Le minimum pour que les symboles et leur action soit claire. Décomposer et pas synthétiser.

Aiguiller et décoder - 1.10

Si on pense à un flux d'information, il faut pouvoir l'aiguiller. Cela se fait avec des multiplexeurs et démultiplexeurs. Sélectionner une position mémoire appelle beaucoup d'autres notions à voir plus loin.

Logique par mémoire morte - 1.11

Pour les fonctions complexes, mettre la table de vérité directement en mémoire, comme on le

Anciennes technologies - 1.11

Il fut une époque où on voyait encore les bits physiquement. Et on utilisait des trucs comme le OU câblé (il faudrait expliquer la logique à transistors avant).

Rebonds de contact - 1.13

Les signaux physiques doivent être mis en forme logique avec des solutions électriques, logiques ou par programmation. Bon exemple d'analyse du monde réel avec un catalogue de solutions.

Comparateur d'égalité - 1.14

L'occasion de suivre des valeurs logiques sur un schéma pour vérifier que ce n'est pas faux, et de se poser la question comment être sûr.

Comparaison multiprécision - 1.15

Intéressant puisque la comparaison peut se faire en partant des poids forts ou faibles. Bonne occasion pour vérifier si on a compris les nombres!

Programmation d'entrées sorties - 1.16

Simple à comprendre et non trivial. Elargit la vision de ce qui se programme au cœur des systèmes

Parité, redondance, fiabilité - 1.17

Ce devrait être le début du chapitre, pas la fin.

Compléments éventuels

Simplification des fonctions logiques (Karnaugh)

Universalité des portes NOR et NAND

Traitement des rebonds de contact par logiciel

Codes correcteurs d'erreur

Notions vues dans le chapitre 2 **Systèmes séquentiels**

Séquentiel, stable, instable, statique et dynamique - 2.1

Quels exemples dans la nature ou le social? Alternance socialistes/démocrates, instabilité dans d'autres pays.

Load statique = révolution Clock dynamique = élections x-annuelles

Bascules bistables - 2.2

Ajouter un démultiplexeur devant une bascule SR est un bon exemple de combinaison de systèmes simples. On voit que le symbole du Latch apporte toute l'information nécessaire et on a compris que le schéma avec des portes ne sera plus jamais utilisé (niveaux d'abstraction).

Diagramme des temps et spécifications - 2.2

Le diagramme des temps est un outil d'analyse essentiel, pas seulement pour montrer l'évolution des signaux électriques. Il clarifie l'évolution des états dans une "machine" et est utile en gestion de projets.

Analyse avant codage ou câblage - 2.4

Il faut par des exemples montrer les représentations qui aident à avoir une idée claire de ce que l'on doit faire. Les organigrammes et diagrammes d'état n'ont pas de forme standard, chaque problème est spécifique, il faut en voir plusieurs pour exercer la pensée computationnelle.

Comprendre que la simultanéité n'existe pas et que cela peut poser problème quand l'électronique est rapide.

Multiplicité des représentations - 2.5

Décortiquer des comportements simples et les observer sous différents angles permet de bien comprendre. Chaque type de problème à ses outils spécifiques pour aider sa maîtrise.

Conception "cas pire" - 2-6

Le "worst case design" est fondamental à comprendre; il explique la fiabilité des systèmes. On peut le concrétiser en disant comment la durée d'un feu piéton devrait être définie: avec le piéton le plus handicapé de Suisse et le chauffeur le plus impatient .

Bascule JK - 2.9

On peut beaucoup jouer à inventer et tester des compteurs. Cela n'apporte pas de nouveaux concepts, mais c'est un jeu utile.

Ordre et nom des bits - 2.10

Utiliser A B C D comme nom n'apporte aucune information. Un compteur binaire, un registre qui transporte une variable binaire ont des sorties qui ont une fonction définie par l'application, et il faut nommer cette fonction le plus clairement possible.

Compteur synchrone et asynchrone - pas traité

Le grand problème avec les Logidules LAMI et LSL, qui a tué les logidules! Il faut l'escamoter. Un exemple simple pourrait montrer la différence. Analogies: horaire cadencé, onde verte.

Décompteur - pas (encore) traité

Un bon moyen pour comprendre les nombres négatifs, le complément à 1 et à 2, en restant au niveau superficiel. C'est dans le chapitre 1 que l'on pourrait expliquer en détail avec l'extension du signe, les conversions de type en C, etc. - contrantes mémoire des anciens langages, sans lien avec les applications

Application: un réveil matin - 2.12, 2.13, 1.14

Les éléments nécessaires sont en place. Le module horloge a en interne un quartz 32 Mhz et une sortie précise à 1Hz. Mais il manque le module Bluetooth qui piloterait l'enceinte acoustique.

Séquenceur - 2.14

La notion de synchronisation d'une action extérieure est fondamentale. Il faudrait l'introduire avec un exemple plus simple.

La problématique des états métastables qui fait que tout système a une probabilité non nulle de planter serait intéressante à présenter.

Compléments éventuels

Double bascule

Comparaison série

Préleveur d'impulsions

Préleveur de périodes

Notions dans le chapitre 3 **Registres et calculatrices**

Ce chapitre le plus important pour la formation est en rédaction et ne montre que quelques schémas et photos de montages qui explicitent bien son contenu futur.

PIPO PISO et compagnie

Concepts intéressants qui seront jugés trop spécialisés

Calculatrice

A expliquer absolument. Le logidule "additionneur série" facilite le câblage et la transformation en générateur de nombres de Fibonacci fait réfléchir à la séquence d'initialisation.

Notions dans le chapitre 4 **Transmissions en série, redondance et cryptage**
(pour 2021 si intérêt)

Compléments éventuels pour mémoire

Machine d'état , câblage, programmation pour une Porte de garage

Hotel Reservation State Transition Diagram (sur le web)

Analyse d'un Feu pour piétons - Câblage (sans délais) et programmation

Compteur en code Gray. Conversion Hexa-Gray en logique câblée/simulée et en programmation

Idées pour des travaux de maturité en informatique

On ne voit jamais du binaire, il existe?

Rappel historique, les premiers ordinateurs étaient décimaux. Gain en mémoire.

Algorithme de conversion. Serial.print et Serial.parse sous Arduino. Python? Android?

Solutions technologiques pour calculer

Pascaline, innovations pour multiplier, diviser.

Règles à calcul et abaquages

Calculatrices à relais, à tube, à transistors: les principes changent?

Si les humains avaient choisi l'octal et pas le décimal, l'informatique se serait développée plus vite?

<https://interstices.info/machines-a-calculer/>

<http://ancmecca.org/les-machines-a-calculer/histoire-du-calcul-mecanique/>

Détection et correction d'erreur

Inventaire des solutions matérielles et logicielles au niveau des bits, des mots, des blocs.

Algorithmes pour tester les mémoires.

Aides à la conception et certification

Diagrammes, organigrammes et normes associées: domaines d'applicabilité et exemples.

Comment sont programmés les logidules?

Programmer le processeur de quelques logidules dans différents langages et différents systèmes/librairies. En particulier, les logidules CNT16, SiPiSo, ALU.

Technologies des mémoires

Solutions multiprocesseurs

jdn 200814