Architecture des système

Fiche

1 Introduction

tab titre

...	...	...

2 Fonctionnement d'un ordinateur

•stockage •traitement
∘ par des circuits simples (sans notion de temps et sans stockage) ∘ par des circuits complexes (avec notions de temps et de stockage) ∘ entre circuits complexes

3 Représentation des valeurs

Bit : unité de stockage de l'information : 0 ou 1
Stockage de valeurs plus complexes : Comment sont stockées ces valeurs?(entiers positifs, entiers naturels, nombres réels, caractères, chaînes de caractère)

• 1 Entiers positifs

on utilise la base en indice
•Comment connaître la valeur décimale d'un nombre écrit dans une base b ?
-soit un nombre binaire (an-1,an-2,...,a2,a1,a0), chaque ai étant écrit dans la base b
-la valeur de (an-1,an-2,...,a2,a1,a0) est sum(i=0..n-1)ai.b^i

•Comment connaître l'écriture dans une base b d'un nombre n écrit en base décimale ?
-écrire les puissances de b
-calculer c et k tel que : – c est compris entre 0 et b-1 – c.$b^k \leq$ n < (c+1).b^k
-retirer du nombre actuel c.b^k : c est le k-ème chiffre de l'écriture de n dans la base b
-recommencer jusqu'à obtenir 0

• Comment passer d'une base b_1 à une base b_2 ?
- passer de la base b_1 à la base 10 - passer de la base 10 à la base b_2
• Comment passer d'une base 16 à une base 2 rapidement ?
- traduire chaque chiffre hexadécimal en 4 bits
• Comment passer d'une base 2 à une base 16 rapidement ?
- lire les bits par blocs de 4, en commençant à droite
- traduire chaque bloc de 4 bits indépendemment (car 16=2^4)

Les Additions :
comme sur les décimaux ; attention aux retenues ;
attention aux débordements de capacité (le résultat peut avoir plus de chiffres que les opérandes)
Ex : faire l'addition de 39 et de 46 en base 2
39 = 1001112 et 46 = 1011102
on obtient bien 10101012 = 85

• 2 Entiers naturels

Pb : il faut un moyen pour stocker les nombres négatifs
Solution initiale : un bit de signe (le plus à gauche)
Ex : représenter 39 et -39 en base 2, avec un bit de signe, sur 8 bits
39 = 100111 ; 39 sur 8 bits : (0)0100111 ; -39 sur 8 bits : (1)0100111

Pb : les additions et les soustractions sont à gérer différemment
0 et -0 ont deux représentations différentes
Solution intermédiaire : complément logique (ou complément à 1)
pour les entiers négatifs, tous les bits sont inversés
Ex: représenter 39 et -39 en base 2, en complément logique, sur 8 bits
39 = 100111 ; 39 sur 8 bits : 00100111 ; -39 sur 8 bits : 11011000

Remarque : les nombres positifs commencent par 0, les nombres négatifs par 1
Comment ajouter deux entiers naturels en complément logique ?
faire l'addition en base 2; si une retenue déborde (dépasse de la capacité prévue), il faut la réajouter au résultat

Problèmes : addition complexe
0 et -0 ont toujours deux représentations différentes
Solution : complément arithmétique (ou complément à 2)
obtenu en ajoutant 1 au complément logique pour les nombres négatifs
Ex: représenter 39 et -39 en base 2, en complément arithmétique sur 8 bits
39 = 100111 ; 39 sur 8 bits : 00100111 ;-39 sur 8 bits : 11011000+1 = 11011001
Comment ajouter deux entiers naturels en complément arithmétique ?
faire une addition en base 2
si une retenue déborde, il faut l'ignorer

• 3 Réels

Pb : il faut un moyen pour représenter les nombres à virgule
Solution : format IEEE 754, simple ou double précision
signe : bit de signe
valeur absolue : mantisse normalisée et exposant (m.2^e), avec 0.5≤m<1 et e entier
Mantisse à bit caché
le premier bit de la mantisse, qui vaut toujours 1, n'est pas stocké
zéro est stocké avec un exposant maximum
Exposant biaisé
l'exposant est décalé d'un biais pour que la valeur stockée soit toujours positive
Rq : simple précision : 1 bit de signe, 23 bits de mantisse, 8 bits d'exposant (biaisé de -126)
double précision : 1 bit de signe, 52 bits de mantisse, 11 bits d'exposant (biaisé de -1022)

Ex : quelle est la valeur décimale de 11000000 00110000 00000000 000000002 ?
bit de signe : 1, donc nombre négatif
exposant : 10000000=128 (valeur biaisée), 128126=2 (valeur non biaisée)
mantisse : 01100...0 (avec bit caché), 101100...0 (sans bit caché) = 1/2+1/8+1/16 = 0,5+0,125+0,0625 = 0,6875
résultat : - 0,6875.2^2 = -2,75

Ex : quel est l'encodage (en simple précision) de 4,25 ?
nombre positif, donc bit de signe : 0
4,25 = 2,125*2^1 = 1,0625*2^2 = 0,53125*2^3, arrêt car mantisse normalisée
exposant : 3 (valeur non biaisé), 3+126=129 (valeur biaisée), donc : 10000001
mantisse : 0,53125 = 0,5 + 0,03125 = 1/2 + 1/32 = 1000100...0 (sans bit caché), 000100...0 (avec bit caché)
résultat : 01000000 10001000 00000000 00000000_2

• 4 Caractères

Comment les caractères sont-ils stockés ?
règles associant chaque caractère à un numéro
exemple : ASCII, UTF-8, UTF-16, ISO 8859-1 (latin1), etc
ASCII = American Standard Code for Information Interchange
table associant chaque caractère (parmi 128) à un numéro sur un octet
exemples : 'A' est 65, 'B' est 66, 'a' est 97, '+' est 43
Propriétés
les caractères minuscules se suivent
les caractères majuscules se suivent
les chiffres se suivent
Problèmes
pas de caractères accentués
beaucoup (33) de caractères non affichables

• 5 Chaînes de caractères

Deux méthodes
stocker le nombre de caractères puis la liste des caractères (approche utilisée en Pascal)
stocker un caractère terminal (approche utilisée en C ou sous DOS)

4 Circuits logiques

Circuit logique

circuit électronique simplifié permettant de calculer une sortie en fonction d'entrées. exemple : allumer une lumière en fonction de la position de plusieurs interrupteurs	Remarque: les entrées et les sorties sont composées de signaux binaires. la manipulation des bits se fait via des fonctions logiques

Fonctions logiques : fonctions de base : NOT, OR, AND ; autres fonctions : XOR, NOR, NAN

Pas A (NOT)

A=0	1
A=1	0

A ou B

	B=0	B=1
A=0	0	1
A=1	1	1

A et B

	B=0	B=1
A=0	0	0
A=1	0	1

A XOR B

	B=0	B=1
A=0	0	0
A=1	0	0

A NOR B

	B=0	B=1
A=0	1	0
A=1	0	0

A NAN B

	B=0	B=1
A=0	1	1
A=1	1	0

Algèbre de Boole
opérations élémentaires : NOT (noté barre), OR (noté +), AND (noté * ou .)
permet de représenter des fonctions logiques, de les simplifier, et de faire du calcul (appelé calcul propositionnel)
Quelques propriétés •$\overline{\overline{A}}$=A •lois de De Morgan : $\overline{A+B}$=$\overline{A}.\overline{B}$ et $\overline{A.B}$=$\overline{A}+\overline{B}$

Table de vérité
liste exhaustive des sorties en fonction de toutes les entrées
Exemple: la lumière L est allumée uniquement quand les interrupteurs A ou B sont ouverts, à moins que l'interrupteur de sécurité S soit ouvert
entrées : A, B, S sortie : L

Exemple tableau et fonction

Synthèse
construction d'un circuit logique à partir d'une fonction logique
Étapes de la synthèse: •construction de la table de vérité •obtention d'une expression logique réalisation du circuit

Exemple Circuit schema
Analyse
construction d'une fonction logique à partir d'un circuit logique
Simplification des fonctions logiques (cf TD)
•par observation de la table de vérité •par propriétés de l'algèbre de Boole •par la méthode des tableaux de Karnaug

5 Circuits séquentiels

Circuit séquentiel
•circuit logique qui intègre une notion de temps et de mémoire •temps : horloge fournissant un signal (à une certaine fréquence) •mémoire : bascule RS

Bascule RS •une entrée R (pour reset) : met la sortie à 0 •une entrée S (pour set) : met la sortie à 1

Exemple tableau fonction logique : Qnew=$\overline{R}$.S+$\overline{R}$.Qold=$\overline{R}$.(S+Qold) =$\overline{R}$+$\overline{\overline{(S+Qold)}}$	Circuit séquentiel de la bascule RS fonction logique : Q=$\overline{\overline{(S+Q)}}$

Exemple Circuit schema :

6 Architecture d'un ordinateur

•Ordinateur
interconnexion de circuits séquentiels + horloge
bus : moyen de communication entre les circuits séquentiels
modèle de Von Neumann

•Types de circuits séquentiels
mémoire
unités de calculs
unités de traitement
périphériques externes

SCHEMA Architecture

Quelques mots-clés

Expression	Definition
•RAM (Random Access Memory)	mémoire principale, en lecture/écriture
•ROM (Read-Only Memory)	mémoire en lecture seule
•BIOS (Basic Input/Output System)	primitives de base de gestion des entrées/sorties, disponibles avant le démarrage du système d'exploitation, sauvegardées en ROM
•CMOS	technologie mémoire historiquement utilisée pour sauvegarder quelques informations du BIOS en mémoire quand l'ordinateur est éteint
•circuit intégré	circuit logique ou séquentiel réalisant une ou plusieurs fonctions
•processeur	composant électronique réalisant des traitements programmables
•microprocesseur	processeur réalisé sur un seul circuit intégré

7 Résumé des principales connaissances à avoir

Résumé des principales connaissances à avoir
Codage : entiers positifs, entiers naturels, réels, caractères
Circuits logiques : synthèse, analyse
Circuits séquentiels : principe
Ordinateur : organisation

8 titre