

Architecture des ordinateurs haute performance

Soraya Zertal

Laboratoire Li-PaRAD, Université de Versailles

E-mail: soraya.zertal@uvsq.fr

Cours 3 : Jeux d'instructions

Architecture et jeu d'instructions DLX

Choix du DLX : Machine “pédagogique” RISC proche des machines telles que MIPS ou PowerPC.

Caractéristiques :

- 32 registres entiers de 32 bits (R0 spécial et toujours à 0)
- 32 registres flottants de 32 bits
- Les instructions Arith. et Log. ne peuvent s'effectuer en mémoire
- Les opérations Arith. et Log. comportent 3 opérandes
- Plusieurs modes d'adressage : immédiat, registres, indexé (déplacement), indirect, absolu

Jeu d'instructions DLX : modes d'adressage

Plusieurs modes d'adressage :

- immédiat :

`add R1,R2,#4; // [R1] := [R2]+4`

- Registres :

`add R1,R2,R3; // [R1] := [R2]+[R3]`

- Indexé (déplacement) :

`Lw R1,18(R2); // [R1] := Memoire[[R2]+18]`

- Absolu :

`Lw R26,896(R0); // [R26] := Memoire[896]`

Jeu d'instructions DLX : classes d'instructions

Pour des raisons de simplicité, on considère les 3 classes d'instructions suivantes (et non R, I, J) :

- ➊ opérations arithmétiques et logiques
- ➋ opérations mémoires (chargement et rangement)
- ➌ opérations de contrôle (branchements avec ou sans condition)

Chaque instruction est codée sur 32 bits avec un opcode de 6 bits (poids fort)

Les données manipulées sont en octets (8bits), demi-mots (16bits), mots (32bits), double-mots (64bits) pour les flottants double précision.

DLX : instructions Arith. et Log.

Format des instructions Arith. et Log. avec et sans valeur immédiate :

Code-op	rs1	rd	immédiat	
6	5	5	5	11

Code-op	rs1	rs2	rd	Fonc.
---------	-----	-----	----	-------

DLX : liste des instructions Arith. et Log.

Instruction	Signification
ADD, ADDU, ADDI, ADDUI	addition (signée et pas), addition immédiate (signée et pas) avec valeur sur 16bits
SUB, SUBU, SUBI, SUBUI	soustraction (signée et pas), soustraction immédiate (signée et pas) avec valeur sur 16bits
Mult, MultU, DIV, DIVU	Multiplication (signée et pas), Division (signée et pas). Reg flottants sur 32bits
OU,OUI,XOR,XORI	ou, ou immédiat, ou-excusif, ou-exclusif immédiat
LHI	chargement Haut immédiat (init haut d'un reg avec val)
SLL, SRL, SRA, SLLI, SRLI,	décalage gauche/droite, logique/arithmetique
S_, S_I	Set if CDT : “_” peut etre LT, GT, LE, GE, EQ, NE

DLX : instructions Arith. et Log.

Quelques exemples :

instruction	fonction	traitement correspondant
Add R1, R2, R3	Addition	$\text{Reg}[R1] := \text{Reg}[R2] + \text{Reg}[R3]$
Addl R1, R2, #5	Addition immédiate	$\text{Reg}[R1] := \text{Reg}[R2] + 5$
SLLI R1, R2, #5	Décalage logique gauche immédiat	$\text{Reg}[R1] := \text{Reg}[R2] \ll 5$
SLT R1 R2 R3	Set if Lower Than	si $(\text{Reg}[R2] < \text{Reg}[R3])$ $\text{Reg}[R1] = 1$, 0 sinon
LHI R1, #13	chargement immédiat haut	$\text{Reg}[R1] := 13 \# \# 0^{16}$

$\# \# 0^{16}$ concaténation avec 16 Zéro à droite.

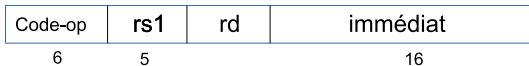
DLX : instructions Arith. et Log.

Bilan :

- 1 Elles opèrent sur des registres, typiquement 2 sources et un destination
- 2 La mise à jour du registre se produit en fin d'exécution
- 3 Leur complexité est modérée
- 4 Leur timing d'exécution est déterministe : une fois les valeurs des registres sources obtenues, la durée d'exécution est fixe.

DLX : instructions chargement/rangement

Format des instructions mémoire ou chargement/rangement de mots, demi-mots et octets.



DLX : les instructions chargement/rangement

Une liste restreinte d'instructions de chargement/rangement :

Instruction	Signification
LB, LBU, SB	chargement octet, chargement octet non signé, rangement octet
LH, LHU, SH	chargement demi-mot, chargement demi-mot non signé, rangement demi-mot
LW, SW	chargement mot, rangement mot (de/vers reg. entiers)

DLX : les instructions chargement/rangement

Il existe aussi :

- Des instructions concernant le chargement/rangement des flottants simple et double précision.
- Des instructions de transfert entre registres spéciaux et registres à usage général.
- Des instructions de transfert de 32 bits de/vers registres flottants vers/de registres entiers.

DLX : instructions chargement/rangement

Quelques exemples :

Instruction	Fonction	Traitement correspondant
LW R1, 30(R2)	Chargement mot	$\text{Reg}[R1] := \text{Memoire}[30 + \text{Reg}[R2]]$
LW R1, 500(R0)	Chargement mot	$\text{Reg}[R1] := \text{Memoire}[500]$
SW 1000(R5), R3	Rangement mot	$\text{Memoire}[1000 + \text{Reg}[R5]] := \text{Reg}[R3]$

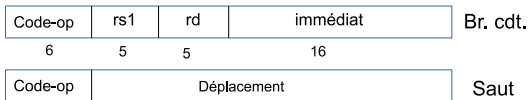
DLX : instructions de chargement/rangement

Bilan :

- ① Leur complexité est importante (calcul d'adresse + accès mémoire), extension et concaténation concernant les transferts entre registres.
- ② Leur timing d'exécution est non déterministe, seul le calcul d'adresse est déterministe. L'accès peut donner lieu à un défaut et engendrer un défaut de plusieurs cycles. Le temps d'accès est donc variable.

DLX : instructions de contrôle

Format des instructions de contrôle : branchement avec (Br. cdt) ou sans condition (saut).



DLX : quelques instructions de contrôle

Les instructions de contrôle les plus utilisées :

Instruction	Signification
BEQZ, BNEZ	branchement si reg égal/différent de zéro au : $CP + \text{val immédiate (16 bits)}$
J, JR	saut au : $CP + \text{déplacement(26 bits)}$ pour J et au Reg[rs1] pour JR

Remarque :

Dans le cas du branchement conditionnel, rs1 est le registre condition, rd n'est pas utilisé.

DLX : instructions de contrôle

Exemples :

Instruction	Fonction	Traitement correspondant
J etiq	Saut	$PC += \text{etiq}$
JR R1	Saut selon registre	$PC = \text{Reg}[R1]$
BEQZ R2 Etiq	Branchement si zéro	if ($\text{Reg}[R2] == 0$) $CP += \text{etiq}$
BENZ R2 Etiq	Branchement si non zéro	if ($\text{Reg}[R2] != 0$) $CP += \text{etiq}$

DLX : instructions de contrôle

Bilan :

- 1 Leur complexité est variable
- 2 Leur timing d'exécution est non déterministe, un branchement peut donner lieu à un défaut et engendrer un défaut de plusieurs cycles