

Arquitectura de Computadores

Grado de Informática

Simulación

Departament d'Enginyeria Informàtica I Matemàtiques

Universitat Rovira i Virgili

Tarragona, Spain



Índice

I. Predictores de Saltos

II. Ejecución y configuración

III. Resultados Estadísticas

Índice

- I. Predictores de Saltos**
- II. Ejecución y configuración**
- III. Resultados Estadísticas**

Predictores de Salto

■ Simulador

sim-bpred, sim-outorder, etc

■ Implementados en los módulos:

bpred.h, *bpred.h*

■ Tipos

- Estáticos: nottaken, taken, perfect,
- Dinámicos: bimod, 2lev(Gshare,Gselect:'Gag',Pag),comb

■ Funciones principales de los módulos:

- bpred_create(class,size)
- bpred_lookup(pred,br_addr)
- bpred_update(pred_addr,targ_addr,result)

Predictores estáticos

- **Basados en información recogida antes de la ejecución del programa.**
- **SimpleScalar implementa:**
 - **Perfect:**
 - Simula el comportamiento del salto. Siempre acierta.
SimpleScalar: -bpred perfect
 - **Nottaken**
 - Retorna constantemente la dirección de la siguiente instrucción.
SimpleScalar: -bpred nottaken
 - **Taken**
 - Predice siempre como tomado.
SimpleScalar: -bpred taken
- **Coste Hardware 0.**
- **Taken obtiene mejores resultados que Nottaken.**

Predictores dinámicos

- **Recogen información dinámicamente.**
- **Los tipos de predictores estudiados:**
 - Predictores de un nivel de historia.
 - Predictores de dos niveles de historia.
 - Predictores híbridos.
- **Simplescalar implementa**
 - one-level (bimodal:bimod)
 - Two-level (2-level : 2lev)
 - híbrido (comb).

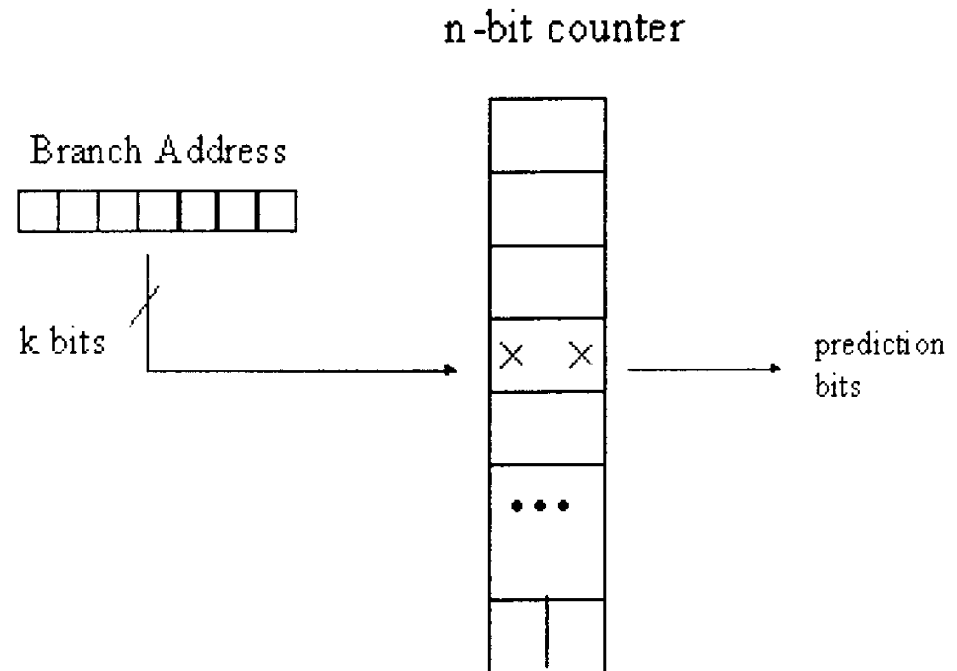
One-level

- **Idea:** k bits de menos peso instrucción de salto para direccionar una tabla 2^k entradas.

- **Entradas:** contador de n-bits saturado.

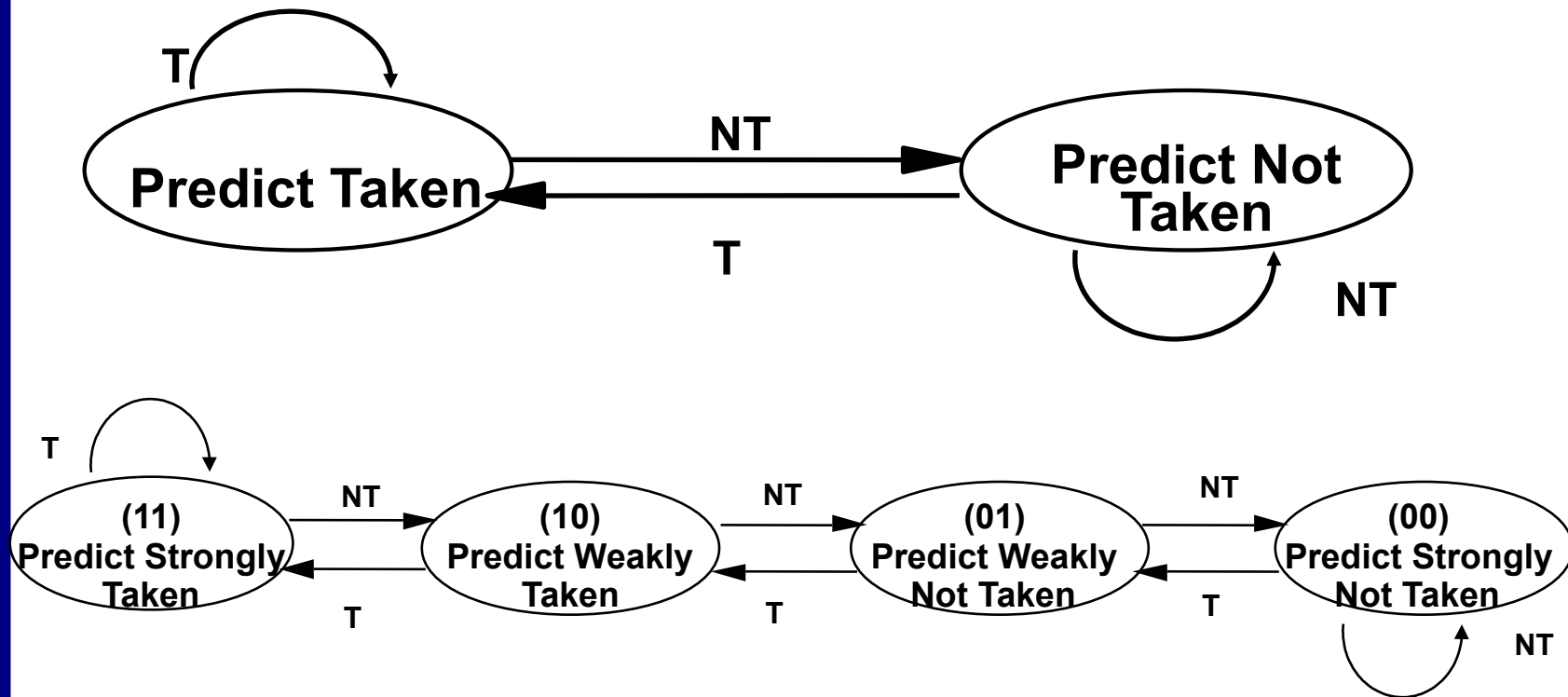
- SimpleScalar:
 - bpred bimod
 - bpred:bimod 2048
 - k=11 y 2048 entradas

One-Level Branch Predictor



One-level

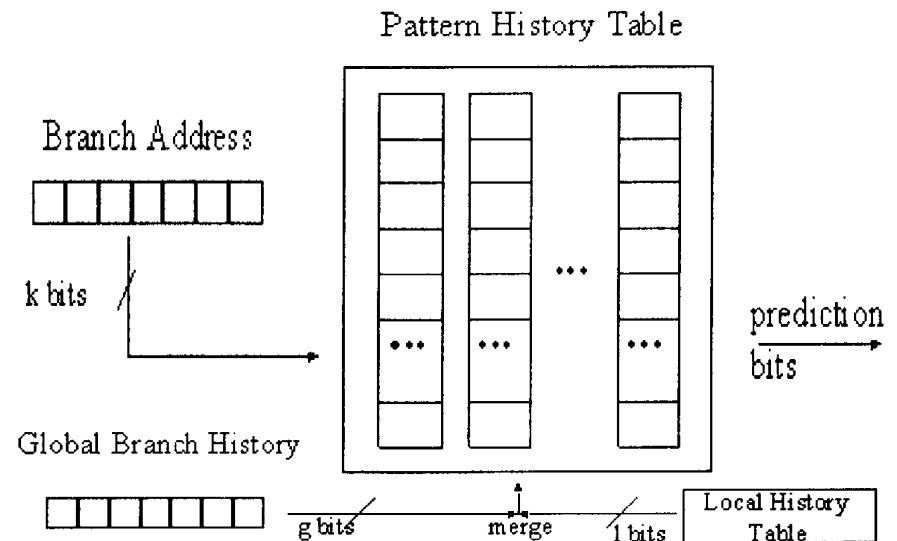
- Contadores Saturados: 1-bit y 2-bits



Two-level

- **Pattern History Table**
- **Branch Address**
- **Global Branch History**
- **Local History Table**
- **Merge**

Two-Level Branch Predictor



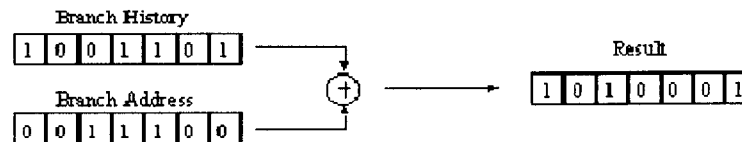
Two-level

Merge (Fusión)

Concatenación.



XOR.



- De forma experimental se ha comprobado que el método de la XOR da mejores resultados que el de la concatenación.

Two-level

- SimpleScalar permite simular varios predictores dentro del 2-level
 - bpred 2lev
 - bpred:2lev <l1_size> <l2_size> <hist_size> <xor>

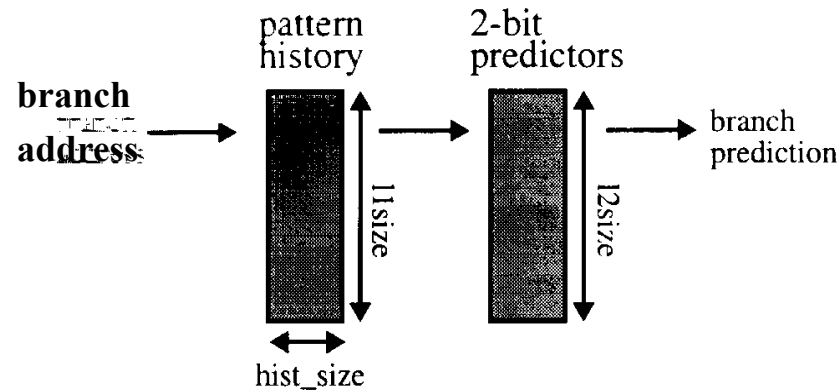


Figure 6. 2-level adaptive predictor structure

	predictor	l1_size	hist_size	l2_size	xor
→	GAg	1	W	2^W	0
	GAp	1	W	$>2^W$	0
→	PAg	N	W	2^W	0
	PAP	N	W	2^{N+W}	0
→	gshare	1	W	2^W	1

Two-level

■ Los valores de configuración permiten simular varios predictores

-bpred:2lev <l1_size> <l2_size> <hist_size> <xor>

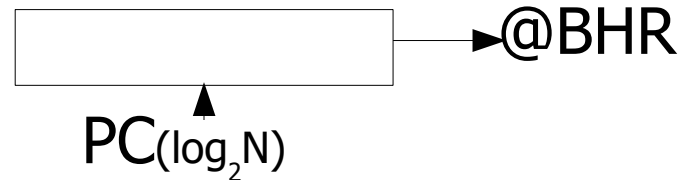
N

M

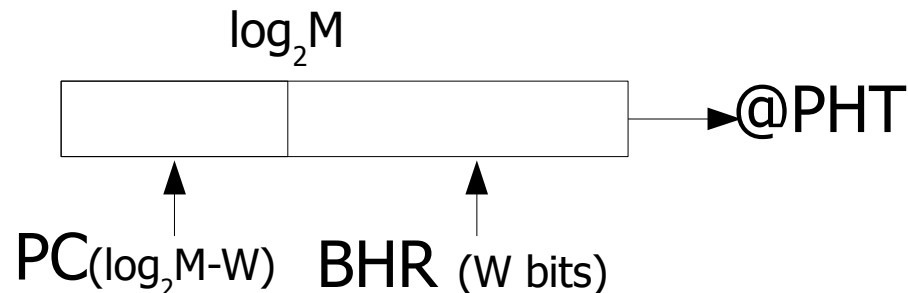
W

X

- Los bits para direccionar al Branch History Registers (N entradas) se cogen del PC ($\log_2 N$ bits) y tiene una anchura de W bits



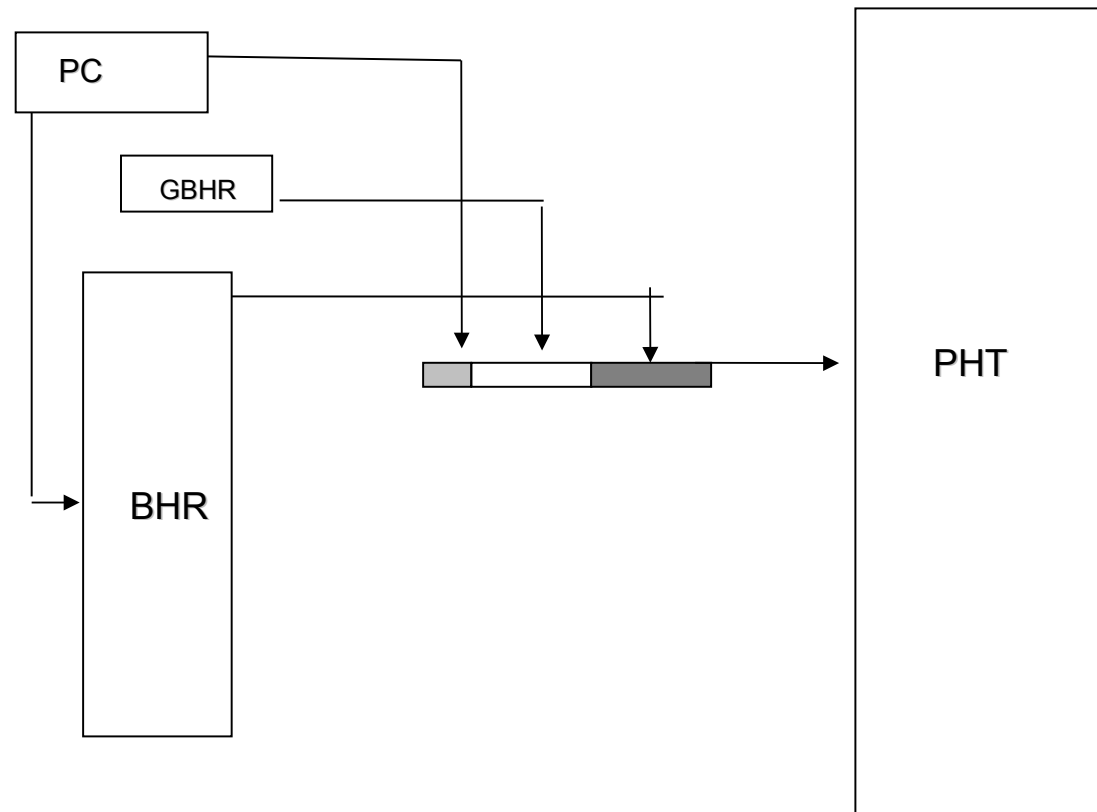
- Los bits para direccionar al Pattern History Table (M entradas) se cogen del PC y de W. De manera que W es menor o igual que $\log_2 M$ bits y los bits que faltan se cogen del PC.



Predictor a implementar

■ Alloyed Branch Predictor:

- De tipo pseudo-híbrido, parecido a un two-level
- Trabaja conjuntamente con historia local y global
- Ataca el problema de la historia incorrecta



Índice

I. Predictores de Saltos

II. Ejecución y configuración

III. Resultados Estadísticas

Ejecutar Simulador

- **El simulador a utilizar es el**
sim-outorder
- **Se modificaran los parametros de configuracion del predictor de saltos**
-bpred,.....
- **Se utilizaran los benchmarks spec2000**

Parámetros del predictor

Parámetro	Argumento	Valor por Defecto
-bpred	<string>	bimod
Predictor a usar:{nottaken taken perfect bimod 2lev comb}		
-bpred:bimod	<int>	2048
En caso de usar un predictor bimodal indica el numero de entradas de la tabla de prediccion de 2-bits saturados		
-bpred:2lev	<int list>	1 1024 8 0
En caso de usar un predictor 2-level indica la configuracion de las tablas: <l1size> <l2size> <hist_size> <xor> L1size : numero de entradas del Branch History Register L2size: número de entradas del Pattern History Table Hist_size: número de bits de historia de los saltos Xor: 0 → concatenación de PC+BHR, 1-> Xor de PC+BHR		

Configuración del predictor

Si queremos un Gshare de 512 entradas de PHT

- Configuración:

-bpred 2lev

-bpred:2lev 1 512 9 1

Si queremos un Gag o Gselect de 1024
entradas de PHT

- Configuración:

-bpred 2lev

-bpred:2lev 1 1024 10 0

Si queremos un Pag de 2048 entradas de PHT y
256 entradas de BHR

- Configuración:

-bpred 2lev

-bpred:2lev 256 2048 11 0

Índice

- I. Predictores de Saltos**
- II. Ejecución y configuración**
- III. Añadir predictor**
- IV. Resultados Estadísticas**

Estadísticas del Simulador

sim_IPC

instructions per cycle

bpred_bimod.lookups	# total number of bpred lookups
bpred_bimod.updates	# total number of updates
bpred_bimod.addr_hits	# total number of address-predicted hits
bpred_bimod.dir_hits	# total number of direction-predicted hits (includes addr-hits)
bpred_bimod.misses	# total number of misses
bpred_bimod.jr_hits	# total number of address-predicted hits for JR's
bpred_bimod.jr_seen	# total number of JR's seen
bpred_bimod.jr_non_ras_hits.PP	# total number of address-predicted hits for non-RAS JR's
bpred_bimod.jr_non_ras_seen.PP	# total number of non-RAS JR's seen
bpred_bimod.bpred_addr_rate	# branch address-prediction rate (i.e., addr-hits/updates)
bpred_bimod.bpred_dir_rate	# branch direction-prediction rate (i.e., all-hits/updates)
bpred_bimod.bpred_jr_rate	# JR address-prediction rate (i.e., JR addr-hits/JRs seen)
bpred_bimod.bpred_jr_non_ras_rate.PP	# non-RAS JR addr-pred rate (ie, non-RAS JR hits/JRs seen)