IDT WinChip™

The Windows Microprocessor

Background

Current Products

The New IDT WinChip 4

IDT WinChip Background

- IDT WinChip C6 Processor First Ships 12/97
  - Socket 7-Compatible (includes MMX™)
  - Developed by Centaur Technology (IDT Subsidiary)
- We Are Targeting The *Value* Market
  - Sub-$1000 Desktops & < $1500 Mobile
- Our Strategy: *The Best Value!*
  - Unique design → smallest die & basic technology → lowest cost → lowest price
  - Good (But Not Best) Performance
- Approximately 500K Sold Since 12/97
- 2nd-Generation WinChip 2 Now Shipping

The IDT WinChip 2

- Significant Performance Improvements per MHz
  - 2x WinChip C6 MMX performance (≈ P55)
  - 2x WinChip C6 FP performance (≈ AMD K6-2)
  - +10% Winstone 98 performance
- Includes 3DNow!™ (AMD Compatible)
  - Significant 3D-graphics performance improvement
- AMD-Compatible 100 MHz Bus (Super7™)
- Low Cost
  - 58 mm^2 in IDT 0.25µ!
- Shipments Started In 9/98
  - Currently 225, 240, 250 & 266 MHz

IDT WinChip Design Philosophy

Goals

- Low Cost (& Low Power) via Smallest Die Size & Use of "Basic" Technology
- "Acceptable" Benchmark Performance
  - Not The Fastest
  - But Fast Enough So Not A Major Buying Factor

Design Approach

- Minimize Bus Stalls → Large Caches & TLBs
  - Large/Fast L1 Better Than Small L1 + Larger L2
- "Simple" Execution Unit → In-Order Execution
- Optimize & Tune For The Target Environment
- Tricks Are Better Than Transistors

IDT WinChip Design Philosophy

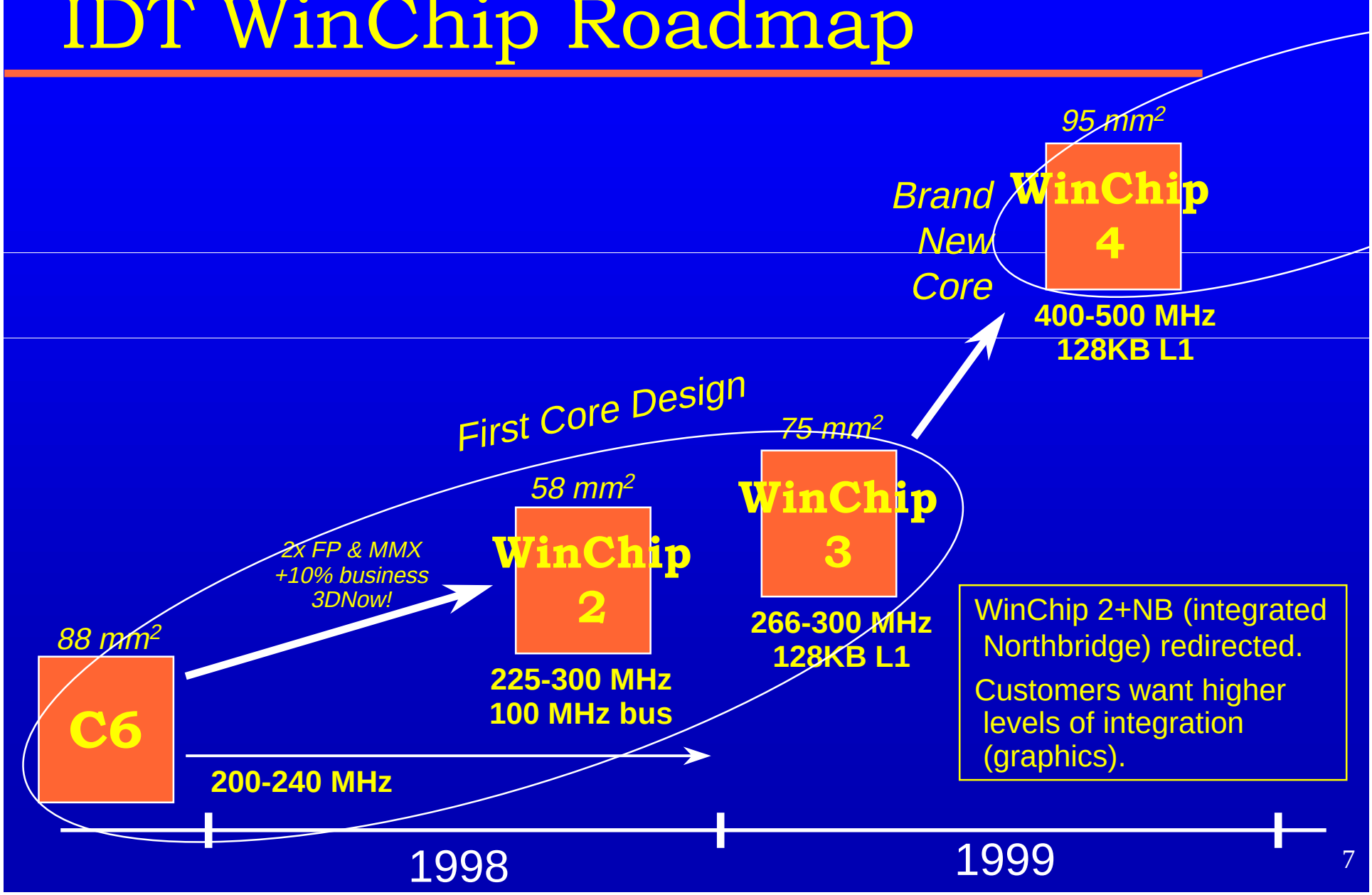
Results

- Low-End PC at 300 MHz (300PR for Cyrix)...

	Cyrix M2	WinChip 2	AMD K6-2	Mendo-cino	Pentium II
size (0.25µ)	88	58 mm^2	81	153	135
WS98 (on Win98)	19.6	20.1	20.9	20.8	21.1
WB98 3D (Riva 128)	354	729 (both use 3DNow! & DirectX 6)	818	860	860

At Same MHz, We're Competitive & Very Low Cost
(But We're Lagging In MHz...)

IDT WinChip Roadmap



WinChip 2+NB (integrated Northbridge) redirected.

Customers want higher levels of integration (graphics).

The New IDT WinChip 4

- New Core Designed Specifically for High MHz
- 500 MHz In Current 0.25µ Technology
  - Faster In Next Year's 0.18µ
- Optimized For Low Cost
  - < 100 mm^2 in 0.25µ ➔ 60 mm^2 in 0.18µ
  - vs. 153 mm^2 For Mendocino (0.25µ)
- Good Benchmark Performance
  - 128 KB L1 Caches (vs. Integrated L2)
  - CPI As Good As Mendocino Would Be At 500 MHz
- "Low" Power (vs. Other 500 MHz Processors)
  - 16W Max At 500 MHz At 2.5V ➔ 9W At 1.8V

Goal The Best Low-End CPU

WinChip 4 Design For MHz

- Microarchitecture...
  - 11-Stage Pipeline (vs. current 6 stages)
  - Pipelined Caches (1-Cycle Throughput)
  - Relatively Simple Control Logic & Datapath
  - Highly Tuned (MHz vs. CPI & Size)
- Highly Custom Layout
  - Including PLAs/Custom for Much Control Logic
- Extensive Use Of Dynamic Logic
- Robust Power & Clocks (Using 6L Metal)
- New Centaur-Developed Tools
  - Automated Stack Placement & Wiring, PLA Generator, Logic Optimization, etc.

WinChip 4 Design For CPI

High MHz Has Major Architecture Implications...

➔ *Deep Pipeline & High Bus Ratio*

➔ *CPI Dominated by Branches & Bus Traffic*

➔ Attack Branch Misprediction

- with Unique State-of-Art Prediction Algorithm

➔ Attack Bus Traffic

- with Large L1 Caches (2x 64KB, 2/4-way)
- with Large L1 TLBs (2x 128, 8-way)
- with Lots of Buffering, Smart Prefetch, Write Allocate, Combining, SmartLock, etc.

➔ Optimized Instruction Execution Capability

- Don’t Waste Transistors Where It Doesn’t Matter
- Especially Considering High Bus Ratios

WinChip 4 Design Comparisons

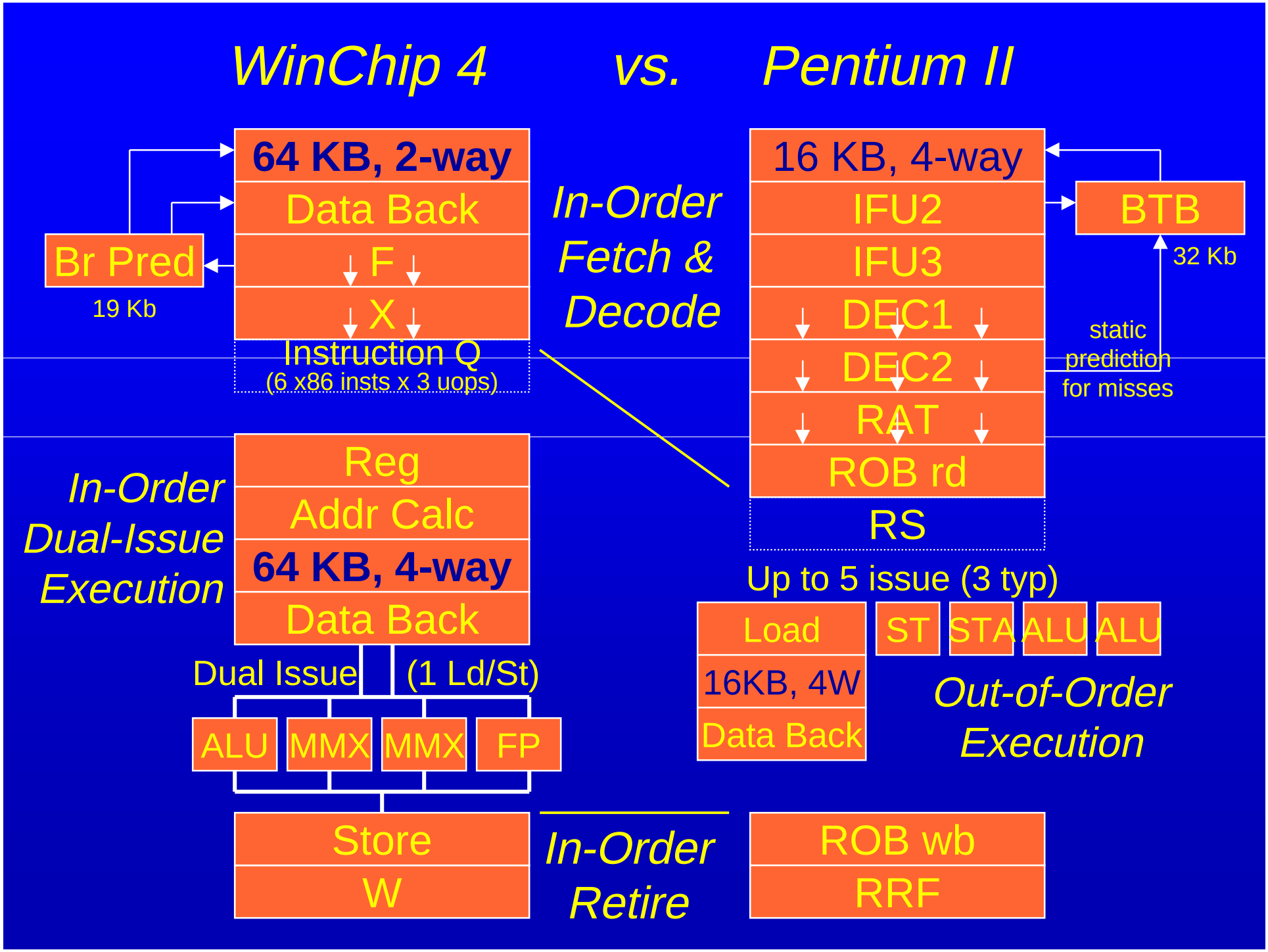
	WinChip 4	AMD K6-3	Mendo-cino	Pentium II
size (0.25μ)	<100 mm²	135	153	131
L1 cache	128 KB 2/4-way	64 2-way	32 4-way	32 4-way
L2 cache (int)	0	256 KB	128 KB	0
L1 TLBs	256 8-way	192 4-way	96 4-way	96 4-way
FS Bus	100 MHz	100	66	100
Branch prediction	advanced	2-level 8k BHT	2-level 512 BTB	2-level 512 BTB
PL Stages	11	6-7	10-12	10-12
Execution	In-order 4 FU's ≤2 issue	O-o-O 6 FU's ≤ 4	O-o-O 5 FU's ≈3	O-o-O 5 FU's ≈3

In-Order vs. Out-of-Order

- Long Bus "Stalls" Can Never Be Eliminated
  - At Bus Ratios of 5x (& Higher)
  - Cache/TLB Misses, I/O, Non-Cacheable (Video)
- Usable ILP Much Less Than Bus Stall Cycles
  - O-o-O Can't "Cover Up" Branch/Bus Stalls
- Out-of-Order Implementation Costs A Lot
  - Pipeline Stages & Transistors ➔ Area & Power
  - That Could Be Used To Reduce Bus Traffic
  - That Could Be Used To Reduce Die Size & Cost
  - That Impact Critical MHz Path
- Out-of-Order Actually Slows Some Sequences
  - vs. Optimized In-Order Design (examples later)

Optimized In-Order Design

- Pipeline Loads Before ALU/Branches
  - Allows 1-Cycle Load-ALU, RETs, Indirect Br, etc.
  - Modern Code Makes Heavy Use of These
- Pipeline Stores After ALU Stages
  - Modern Code Makes Heavy Use of Load-ALU-Store
  - Facilitates Store-Load Forwarding
- Many Specific Code Sequence Optimizations
  - Very Efficient for Highly Used x86 Sequences
- Dual Execute (Only) High Leverage Pairs
  - MMX & 3DNow! Code Highly Pairable
  - Certain Integer Sequences Highly Pairable
  - More Than Dual Execute Has Minimal Gain


WinChip 4
vs.
Pentium II
64 KB, 2-way
Data Back
F
X
Instruction Q
(6 x86 insts x 3 uops)
Br Pred
19 Kb
In-Order
Fetch &
Decode
16 KB, 4-way
IFU2
IFU3
DEC1
DEC2
RAT
ROB rd
RS
BTB
32 Kb
static
prediction
for misses
In-Order
Dual-Issue
Execution
Reg
Addr Calc
64 KB, 4-way
Data Back
Dual Issue
(1 Ld/St)
ALU
MMX
MMX
FP
Up to 5 issue (3 typ)
Load
16KB, 4W
Data Back
ST
STA
ALU
ALU
Out-of-Order
Execution
Store
W
In-Order
Retire
ROB wb
RRF

Code Example

- Real Code From Compiler (unfortunately)

```
add    count,1          ; count is mem variable
cmp    count,8
(jl    loop)
```

- WinChip 4 Generates 2 µops Taking 2 Clocks

```
ld-add-st    count,1  ; store result forwarded
ld-cmp       count,8  ;  directly into ld-cmp
```

- Pentium II Generates 6 µops Taking 5 Clocks

```
load          st addr
(load 2)      load
(load 3)      (load 2)
add           (load 3)
store         cmp
```

But, other functional units are available to do "subsequent" instructions during this period

- Plus, Pentium II 3x As Likely To Miss L1 Cache!

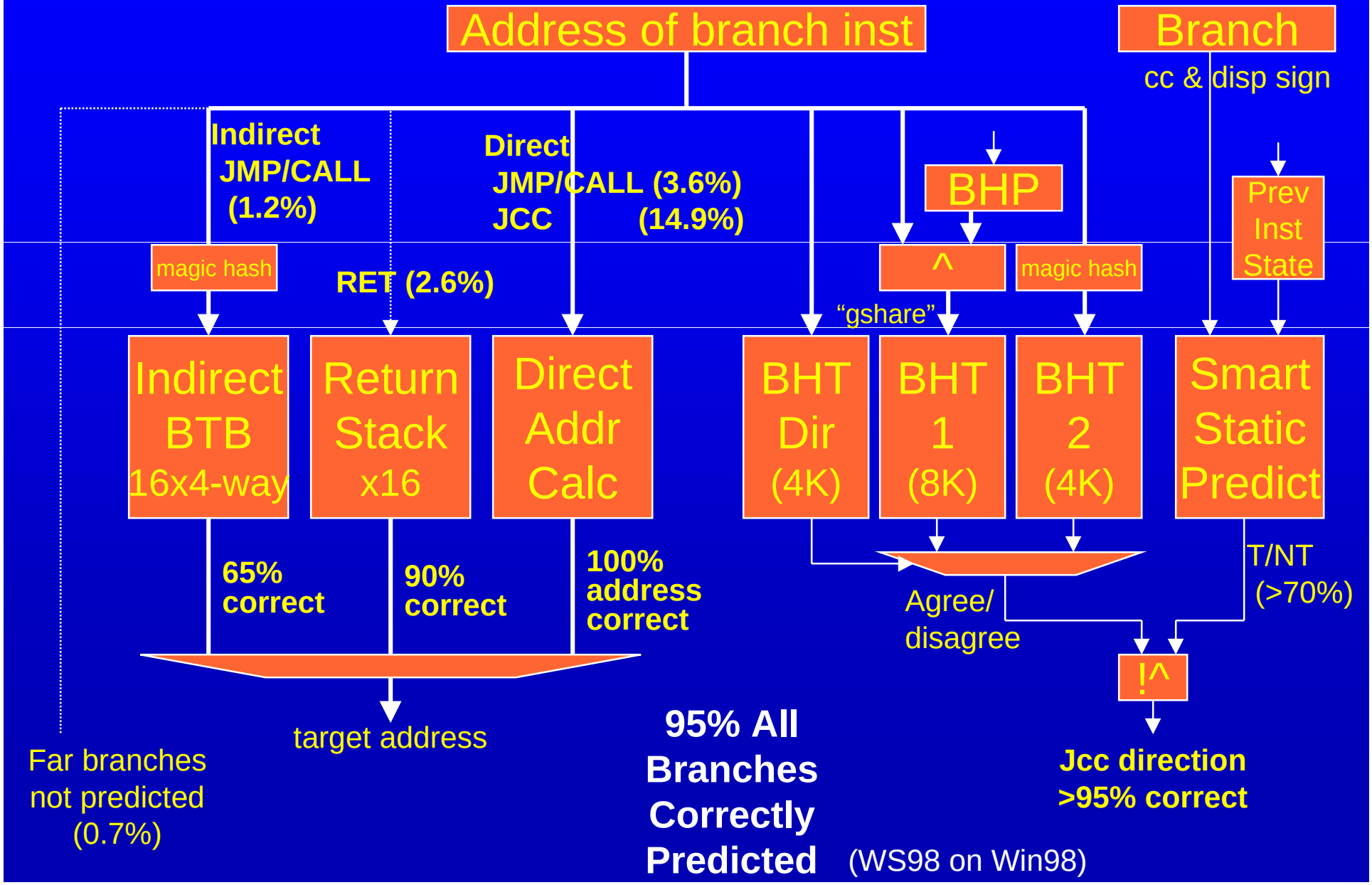
WinChip 4 Branch Prediction

- 3 Mechanisms To Get Branch Address
  - Direct Calculation of IP-displacement forms
  - LIFO Call/Return Stack
  - Small BTB for Indirect Branches
- Two Alternate Predictors of Jcc Direction
  - "gshare" with 8K Entries (13-bit Global History)
    - works well for context-sensitive branches
    - used on WinChip 2 (4K entries)
  - 4K Single-Level Predictor
    - works well with highly-predictable branches
  - Both Use 1-bit "Agree/Disagree" Entries
    - with sophisticated static prediction mechanism
- Jcc Predictor Selector Table
  - Selects Which Predictor Based On Previous History

19.5 Kbits
vs.
32 Kbits for Pentium II

same hit %

WinChip 4 Branch Prediction



The New IDT WinChip 4

- New Low-Cost High-MHz Core Design
  - Highly Optimized for MHz
  - 128 KB L1 caches, Great Branch Prediction, Highly Optimized In-Order Execution
- Version 1
  - 2H99 Production
  - Socket 7
  - 0.25µ (6L Metal), 2.5V (<100 mm^2)
  - 400–500 MHz
- Version 2
  - 1H00 Production
  - More MHz, Better CPI, Different Bus/Integration
  - 0.18µ, 1.8V (60 mm^2)
  - 500–700 MHz