Rise of the Virtual Machines

32 years of READY prompts

By Darek Mihocka, Emulators.com

Originally presented at Conestoga College
on February 7, 2012, posted online Feb. 11

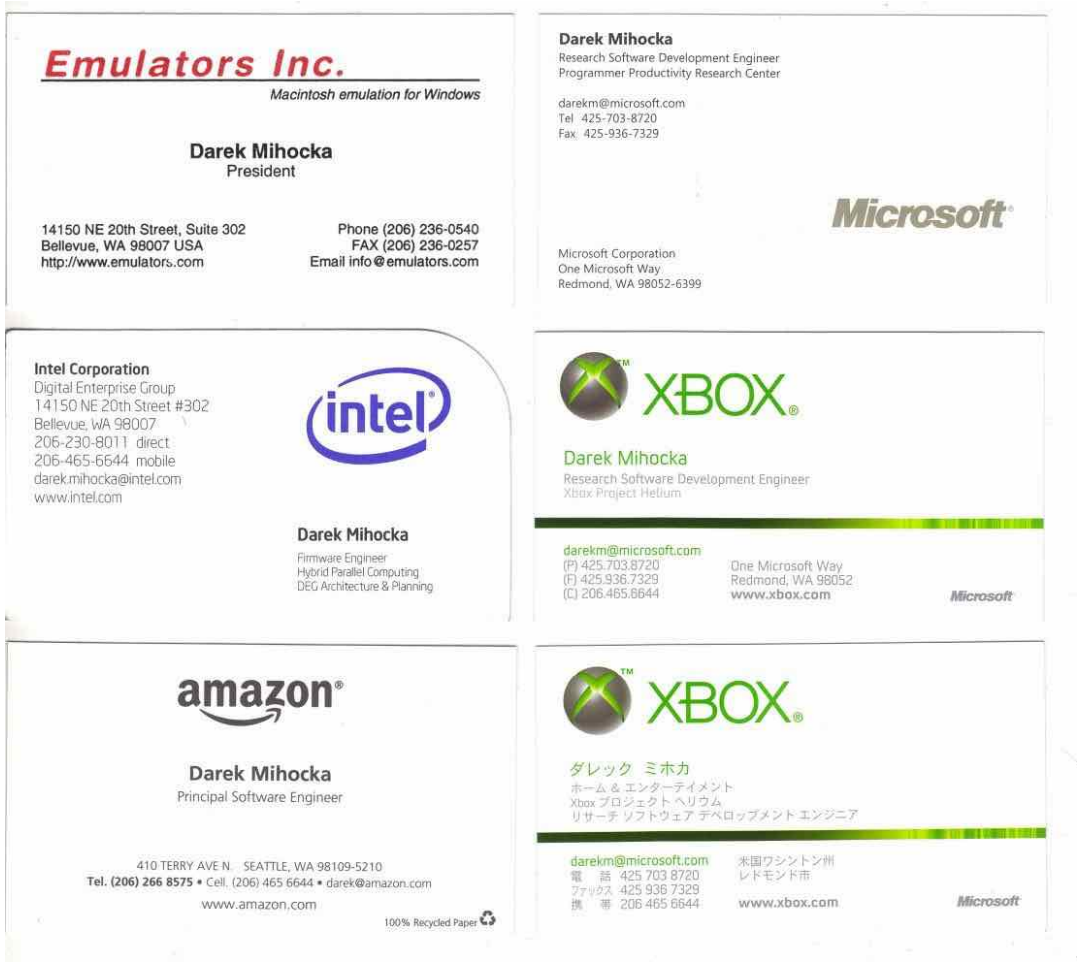
And I am…?

- Grew up in Toronto, degree from Waterloo
- 1980 – first used Commodore PET at school
- 1981 – purchased my first Atari 400
- 1985 – published first magazine article
  - software primitives in 6502 assembly language
  - http://www.atarimagazines.com/v4n2/GUP.html
  - Also started Comp Engineering at U of Waterloo
- 1986 – worked with Ignac (your Ignac) on co-op
  - That first day of work Chernobyl melted down, oops

More Recently…

- Worked at a few little startups and projects over past years:

- 1988 – Brasoft
- 1990 – Microsoft
- 1997 – Emulators
- 2001 – MSR
- 2005 – Xbox 360
- 2008 – Intel
- 2010 – Amazon EC2



Piracy Made Us Do It

- Summer of 1986 Ignac and I each bought at Atari 1040ST computer and monitor
  - Similar to more expensive Apple Macintosh
  - About $1500 for ST + monitor, floppy disk only!
- We had all this nice pirated Apple II and Atari 400/800 software that didn't run on the ST
- Too naïve to know better, set upon to write a 6502 and Atari II emulator on the Atari ST
  - Even Atari developers told us couldn't be done
  - Started coding 6502 interpreter in C in August

ST Xformer is Born

- Apple II READY prompt booted up in October 1986, Atari 800 READY prompt by end of year
- COMPUTE! magazine rejected ST Xformer
  - saying none of their readers would care about it
- Posted to CompuServe in January 1987
  - lots of people cared and downloaded it!
- Neither Apple nor Atari were pleased
- Lawyers called, threatened to sue because of copyright infringement of their ROMs

Darek vs. Atari

- I fought back by publicizing the threats
- The Atari user community stepped in
- John Nagy of **Computer Shopper** magazine wrote a piece in June 1987 issue about big bad Atari twisting the arm of little college boy
- Atari and I reach agreement to open source it:
  - http://www.atariarchives.org/cfn/05/11/0050.php
- Published in **ST LOG** magazine later in 1987
  - http://www.atarimagazines.com/st-log/issue26/18_1_INSIDE_ST_XFORMER_II.php

Life at Microsoft

- January 1987 I took 4-month co-op internship at little startup called Microsoft in Redmond, suburb of Seattle
- How I got the gig: came in to walk-in interview on campus carrying a 100 page printout of ST Xformer source code
- Got lost getting to my first day of work
  - Redmond was farmland and forest back then
- 900 employees in 4 buildings developing products for Apple II, Atari ST, Xenix, Macintosh, DOS, OS/2, and Windows
- Even met Bill Gates before he was a billionaire, woo!
- First gig – porting Multiplan to Japanese
  - Development machine: 8 MHz IBM AT running OS/2 1.0
- Made enough to buy 20 MB Atari SH204 hard drive

The Cost of Storage

- 1987 - $1000 buys 20 MB Atari hard disk
- 2012 - $200 buys 2 TB hard disk
- 100,000x the storage for 1/5th the price!
  - Bang for the buck improvement of 500,000x

A Business is Born

- 1988 we founded Branch Always Software
- “Branch Always” is the 68000 unconditional branch instruction at the start of every Atari executable file and ROM, like, duh
  - Ok, it was far too geeky, nobody understood the name
- ST Xformer II rewritten and optimized in 6502 assembly emulates Atari 400/800XL/130XE
  - Atari OS and BASIC used with permission. (C) 1979-1984 Atari Corp.
- We had our first shareware product!

Quick Tools

- By 1990 developed the “Quick ST” software accelerator, “Quick CLI” command line prompt, and dozen other useful tools we bundled into a package called “Quick Tools”
- Sold at Atari computer shows and swap meets in places like Detroit, Toronto, Pittsburgh, Seattle
- Also advertised in magazine ads
- Lots of daily cheque deposits and late night duplicating floppy disks and shrink wrapping
    - Online distribution had yet to be invented!

To The Evil Empire I Go!

- July 1990 I started full time “blue badge” at Microsoft, working on the Windows 3.0 port of Microsoft Works
- Branch Always Software U.S. branch opens
  - 14150 N.E. 20th Street, Suite 302, Bellevue, WA 98007
  - Still the official business address of **www.Emulators.com**
- Worked 8 hours a day at Microsoft, 4 to 6 hours a night on Brasoft for next 7 years

Gemulator is Born

- Early 1991 I do some prototyping and determine that full speed 8 MHz MC68000 emulation is possible on 33 MHz 80386
- "**GEMULATOR**" is born
- 18 months of coding later, Gemulator 1.0 is released at the Glendale Atarifest Sep. 1992
  - So began a steady pattern of leaving Microsoft at 5pm Friday, packing up the car, driving all night, exhibiting at Atari show on Saturday, driving all night back, sleeping all day Sunday, back to work on Monday

Brasoftmobile

- 1991 Saturn SL-2 became the Brasoftmobile which I drove to Atari events in San Jose, Glendale, Dallas, Vancouver, Chicago, Portland

Meanwhile at the day job…

- 1992 switched over to DevDiv to work on the Apple Macintosh compiler tools for 68040 and the then new PowerPC Macs
- "**Microsoft Visual Studio Cross-Compiler Edition for Macintosh**" actual retail product!
- Wrote 20000 lines of PowerPC assembly to implement P-code interpreter and debugger
- Even had to write my own PowerPC assembler
- A decade later, these tools used for Xbox 360!

Why P-code?

- Little known fact that many Microsoft products for MS-DOS, 16-bit Windows, and Macintosh were compiled to P-code bytecode.
- Retail compilers even supported -Oq switch
- P-code delivered 2x code size reduction over native x86 and 68040 code, 3x for PowerPC
- Improved boot times!
  - Fewer disk sectors read, lower memory footprint
- P-code bytecode became MSIL bytecode for .NET
  - Today's Visual Studio compilers use -clr instead of -Oq

Time Travel Debugging is Born

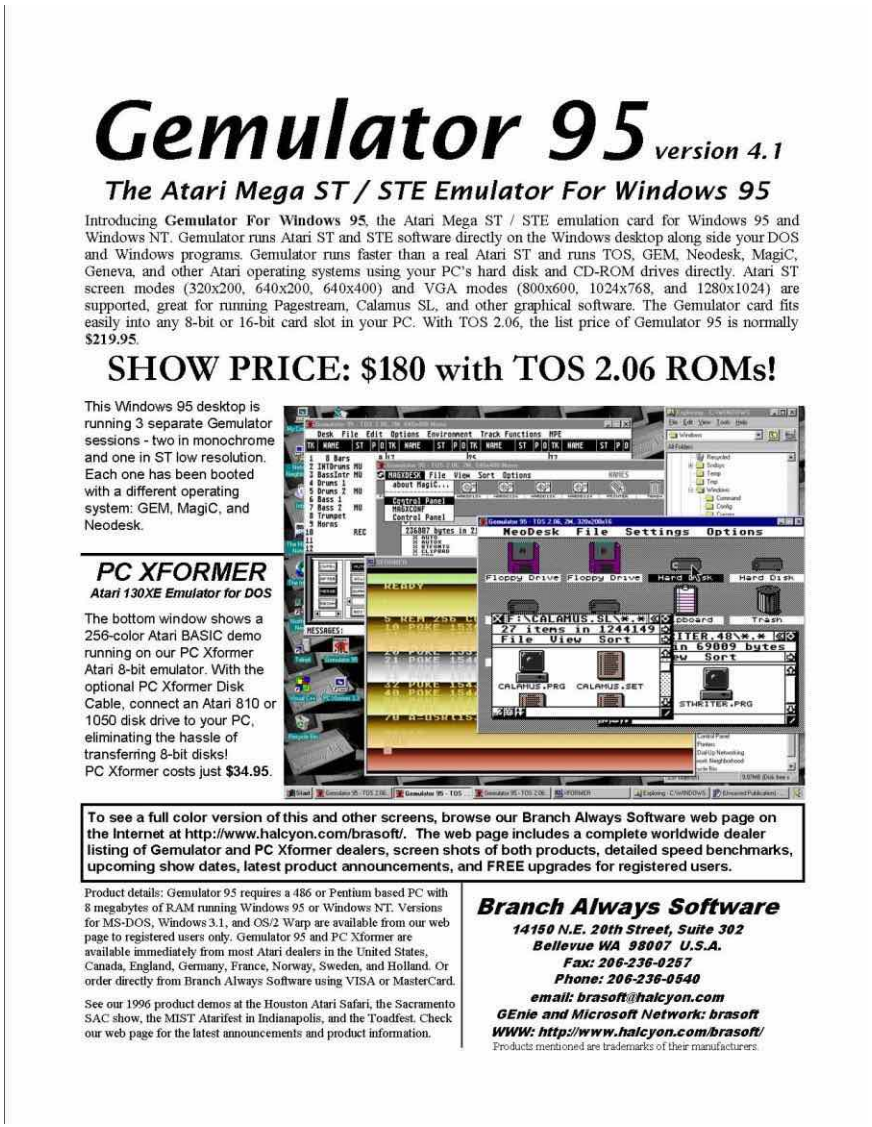
- One useful benefit of the P-code interpreter is that it could record machine state at every single instruction interpreted
- Useful to step backwards from a crash
  - Say, to step back from bad pointer access to the point where the pointer got corrupted
- Easier to debug crashes using P-code than the native build and native PowerPC debugger

The Boundless Energy of Youth

- Between 1994 and 1997 worked on 32-bit Visual Studio compiler, 68040 compiler, PowerPC compiler, linker, debugger, assembler, 32-bit Office 95 port, Unicode Office 97 port, and PowerPC Mac Office 98
- All while also developing new MS-DOS and Windows versions of Gemulator and promoting it on weekend at Atari events
- Also bought two condos and a house (which is now filled with over 50 Macs, PCs, and Ataris)

Seriously…


Works
Version 2.0
September 29, 1991
Visual C++
Version 5.0
February 12, 1997
VISUAL C++
Version 2.0
September 21, 1994
Visual Studio 97
Version 5.0
February 12, 1997
WORD
for the Macintosh
Version 6.0
August 8, 1994
DAREK MIHOCKA
Word
Designed for Windows 95
Version 7.0
July 24, 1995
Windows xp
Service Pack 2
August 6, 2004
Word 97
Version 8.0
November 18, 1996
XBOX 360
November 22, 2005
Office 97
November 18, 1996
Xbox Project Helium
June 15, 2006


Gemulator 95 version 4.1
The Atari Mega ST / STE Emulator For Windows 95
Introducing Gemulator For Windows 95, the Atari Mega ST / STE emulation card for Windows 95 and Windows NT. Gemulator runs Atari ST and STE software directly on the Windows desktop along side your DOS and Windows programs. Gemulator runs faster than a real Atari ST and runs TOS, GEM, Neodesk, MagiC, Geneva, and other Atari operating systems using your PC's hard disk and CD-ROM drives directly. Atari ST screen modes (320x200, 640x200, 640x400) and VGA modes (800x600, 1024x768, and 1280x1024) are supported, great for running Pagestream, Calamus SL, and other graphical software. The Gemulator card fits easily into any 8-bit or 16-bit card slot in your PC. With TOS 2.06, the list price of Gemulator 95 is normally $219.95.
SHOW PRICE: $180 with TOS 2.06 ROMs!
This Windows 95 desktop is running 3 separate Gemulator sessions - two in monochrome and one in ST low resolution. Each one has been booted with a different operating system: GEM, MagiC, and Neodesk.
PC XFORMER
Atari 130XE Emulator for DOS
The bottom window shows a 256-color Atari BASIC demo running on our PC Xformer Atari 8-bit emulator. With the optional PC Xformer Disk Cable, connect an Atari 810 or 1050 disk drive to your PC, eliminating the hassle of transferring 8-bit disks! PC Xformer costs just $34.95.
NeoDesk File Settings Options
To see a full color version of this and other screens, browse our Branch Always Software web page on the Internet at http://www.halcyon.com/brasoft/. The web page includes a complete worldwide dealer listing of Gemulator and PC Xformer dealers, screen shots of both products, detailed speed benchmarks, upcoming show dates, latest product announcements, and FREE upgrades for registered users.
Product details: Gemulator 95 requires a 486 or Pentium based PC with 8 megabytes of RAM running Windows 95 or Windows NT. Versions for MS-DOS, Windows 3.1, and OS/2 Warp are available from our web page to registered users only. Gemulator 95 and PC Xformer are available immediately from most Atari dealers in the United States, Canada, England, Germany, France, Norway, Sweden, and Holland. Or order directly from Branch Always Software using VISA or MasterCard.
See our 1996 product demos at the Houston Atari Safari, the Sacramento SAC show, the MIST Atarifest in Indianapolis, and the Toadfest. Check our web page for the latest announcements and product information.
Branch Always Software
14150 N.E. 20th Street, Suite 302
Bellevue WA 98007 U.S.A.
Fax: 206-236-0257
Phone: 206-236-0540
email: brasoft@halcyon.com
GEnie and Microsoft Network: brasoft
WWW: http://www.halcyon.com/brasoft/
Products mentioned are trademarks of their manufacturers.

Leaving Microsoft

- 1997 – Office 97 out the door, Mac Office 98 was almost ready to RTM
- Decided to not stay for yet another round of “adding Clippy” and instead focus full time on Gemulator as a business
- Mac Office 98 forced Macintosh users to upgrade to PowerPC machine, I saw an alternate… emulate Mac apps on Windows
- Sep 2 1997 – handed in my badge, age 30

Proving Them Wrong Again

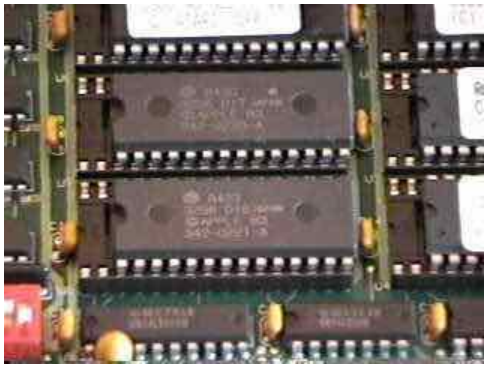
- They said you can't possibly emulate a Macintosh on a PC – it would be too slow
- Except I'd already done it ☺
- October 1997 – I demoed prototype Mac Plus emulator at a user group swap meet in Dallas

Taking the Plunge

- After Dallas, purchased booth space at Macworld 1998 in San Francisco
- Problem is, I had no product! The prototype I showed in Dallas was based on running Dave Small's Mac emulator on top of Gemulator
- Also the slight problem of **Apple's lawyers**
- Fortunately, both Atari ST and Apple Macintosh (Mac Plus, SE, Classic, all II models) used similar ROMs – just need to read them on PC somehow

Fortunate Decision

- Back in 1992 I didn't want Atari to get ticked off and sue over the Atari ST emulator either
- Decided to make Gemulator use real Atari ST ROMs taken from real machines or purchased from Atari resellers – beaucoup $$$ for Atari
- ROM cards allowed reading of both Atari ROMs as well as Apple Macintosh ROMs

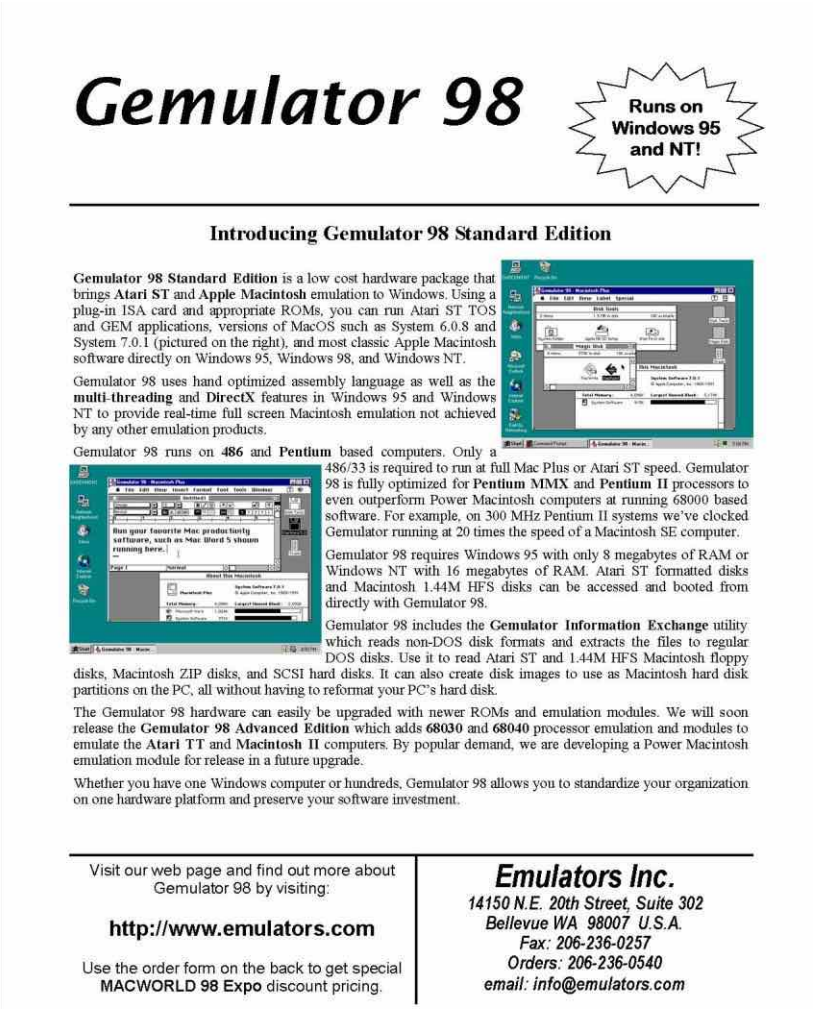


Designing a ROM Card

- Super simple design costing $20 a card to manufacture
- Used the 8-bit DIP switch in lower corner and an 8-bit comparator to dial in the I/O port of the ISA card
- Five 4-bit counters along edge make 20-bit counter, sufficient to index 1 megabyte of ROM
- Eight 32-bit sockets, compatible with 27010 and smaller ROMs, sufficient to read all existing Atari ST/STe/TT and Mac Plus/SE/II/LC ROMs
- Two chips to buffer the output data
- One chip to do chip select of one of the 8 sockets
- I have some of these cards left if you need one!

Macworld 98

- By January 1998 I had implemented Apple Mac Plus hardware emulation on top of Gemulator and was booting System 6.0.8
- Gemulator 98 now emulated Atari 800, Atari ST, and Apple Macintosh
- Printed up a few thousand flyers and headed off to Macworld 98



Gemulator 98

Runs on Windows 95 and NT!

Introducing Gemulator 98 Standard Edition

Gemulator 98 Standard Edition is a low cost hardware package that brings **Atari ST** and **Apple Macintosh** emulation to Windows. Using a plug-in ISA card and appropriate ROMs, you can run Atari ST TOS and GEM applications, versions of MacOS such as System 6.0.8 and System 7.0.1 (pictured on the right), and most classic Apple Macintosh software directly on Windows 95, Windows 98, and Windows NT.

Gemulator 98 uses hand optimized assembly language as well as the **multi-threading** and **DirectX** features in Windows 95 and Windows NT to provide real-time full screen Macintosh emulation not achieved by any other emulation products.

Gemulator 98 runs on **486** and **Pentium** based computers. Only a 486/33 is required to run at full Mac Plus or Atari ST speed. Gemulator 98 is fully optimized for **Pentium MMX** and **Pentium II** processors to even outperform Power Macintosh computers at running 68000 based software. For example, on 300 MHz Pentium II systems we've clocked Gemulator running at 20 times the speed of a Macintosh SE computer.

Gemulator 98 requires Windows 95 with only 8 megabytes of RAM or Windows NT with 16 megabytes of RAM. Atari ST formatted disks and Macintosh 1.44M HFS disks can be accessed and booted from directly with Gemulator 98.

Gemulator 98 includes the **Gemulator Information Exchange** utility which reads non-DOS disk formats and extracts the files to regular DOS disks. Use it to read Atari ST and 1.44M HFS Macintosh floppy disks, Macintosh ZIP disks, and SCSI hard disks. It can also create disk images to use as Macintosh hard disk partitions on the PC, all without having to reformat your PC's hard disk.

The Gemulator 98 hardware can easily be upgraded with newer ROMs and emulation modules. We will soon release the **Gemulator 98 Advanced Edition** which adds **68030** and **68040** processor emulation and modules to emulate the **Atari TT** and **Macintosh II** computers. By popular demand, we are developing a Power Macintosh emulation module for release in a future upgrade.

Whether you have one Windows computer or hundreds, Gemulator 98 allows you to standardize your organization on one hardware platform and preserve your software investment.

Visit our web page and find out more about Gemulator 98 by visiting:

http://www.emulators.com

Use the order form on the back to get special **MACWORLD 98 Expo** discount pricing.

Emulators Inc.
14150 N.E. 20th Street, Suite 302
Bellevue WA 98007 U.S.A.
Fax: 206-236-0257
Orders: 206-236-0540
email: info@emulators.com

SoftMac is Born

- Main feedback from Macworld 98 was:
  - Make it run Mac OS 8, not just System 6 or 7
  - Make it run in colour, not just monochrome
  - Make it emulate Mac II and Quadra, 68030/68040
- Spent rest of 1998 implementing all that
- Rebranded “Gemulator” as “**SoftMac**”
- Returned to Macworld in 1999 for both San Francisco and New York shows, even Tokyo in 2001, and also COMDEX Las Vegas 1999

Mac Users Love It

2000 – A Great Year

- In 2000 bought rights to competing Mac emulator for MS-DOS called "**Fusion PC**" and added to product lineup:
  - http://emulators.com/fusion.htm

- Open source vMac emulator adds Gemulator ROM card support
- Running System 7 and Mac OS 8 now possible on MS-DOS, Windows, Linux. **Emulating Macintosh on PC a success!**
  - As Dilbert says: *they* were idiots!

- My tuning of 68040 emulation code led me to expose performance anomalies in the Intel Pentium 4 processor which I blogged about in December 2000, to unexpectedly find myself fueling the backlash against the Pentium 4:
  - http://www.theregister.co.uk/2001/01/10/pentium_4_high_risk_strategy/

2001 - Oops

- By 2001 Apple had moved on to Mac OS X, new Mac models, and stopped supporting 68040, Mac OS 8, or any of the old machines
- I was already prototyping and demoing a new PowerPC emulator to run emulate the new Mac systems
  - http://www.macworld.com/article/18287/2001/07/emulation.html
  - Had reverse engineered PowerMac 6100 hardware using 160-channel logic probe and capturing traces of 1 million clock cycles at a time
  - Then comparing logic traces with emulator traces to find bugs in emulator where the traces diverged
- Shortly after Macworld New York, 9/11 happened
  - Economy tanked, orders dried up overnight
  - I decided to get a day job again

Back to Microsoft

- October 2001, as Windows XP and XBOX were being released, I returned to Microsoft to go work in MSR (Microsoft Research) and apply my emulation skills at real world problems

- The problems at hand – performance problems on Pentium 4, Windows instability, programs crashing, profiling managed .NET applications – and no tools to tackle with

Nirvana is Born

- “Nirvana” is codename for a project to dynamically instrument Windows applications
- Much like running a P-code interpreter or an emulator, the idea is to be able to trace and record program execution of every thread of Windows application to figure out what the Windows application is doing.
- The “old” common way is static instrumentation
  - Recompile/relink a program with extra instrumentation
  - This ***only*** works with native x86 code, and only if you have source code and tools to rebuild the executable.
- What about 3rd party code? What about .NET/Java?

Developing Nirvana

- From 2001 to 2005 I worked on developing the x86 instrumentation engine
- First started as a plain simple x86 interpreter that emulated Ring 3 user mode instructions, including x87, MMX, SSE by breaking them down into simpler CPU-like micro-ops:
  - e.g. load register, load memory byte, add registers
  - Both Xformer and Gemulator had used similar technique
- By 2003 added a jitter to take the decoded micro-ops and jit them into simple sequences of instructions
- By 2004, added a C/C++ API to register callbacks on events such as calls, branches, memory loads/stores

It Works!

- Nirvana team demos at 2004 MSR “Techfest”
- 5th most popular exhibit – let’s ship a product!

iDNA is Born

- In late 2004 the Nirvana framework released internally at Microsoft.
- In 2005, 64-bit support and a trace/playback API called “iDNA” were added for Nirvana 2.0 release to simplify use of Nirvana
- Time Travel Debugging of Windows apps becomes a reality!
- In 2006 Nirvana paper published at VEE06 at PLDI 2006 in Ottawa:
  - http://www.cs.purdue.edu/homes/xyzhang/fall07/Papers/wenke.pdf
- Office 2007, Windows 7, Internet Explorer, .NET all used Nirvana and iDNA to track down difficult crashes, memory leaks, performance issues
  - Unfortunately not in time to save bloated Vista ☹

iDNA Inspires Intel

- Microsoft was granted U.S. Patent #7,620,938 for Nirvana's iDNA trace/playback mechanism, which used a novel technique to shrink the trace entropy to **under 1 bit per instruction** traced
- Intel later developed PinPlay to mimic iDNA
  - PinPlay wins best paper at CGO 2010 in Toronto
- Intel has now evolved it to full-system ZSIM:
  - http://www.cs.virginia.edu/kim/docs/wish11zsim.pdf

Nirvana's Legacy

- Nirvana and iDNA showed the world that it is possible to trace a Windows application:
  - At every instruction on every thread
  - At only 5x to 15x slowdown (including capturing and writing that trace data to disk)
  - To replay the trace data in a Time Travel Debugger
  - To find memory leaks and other difficult bugs
- PIN, Valgrind, DynamoRIO, QEMU, and Bochs all exist today to similarly help developers

Emulation in the Xbox 360

- In late 2005 Microsoft launched the Xbox 360
- XBOX used variant of Intel Pentium III
- Xbox 360 used in-order variant of PowerPC G5
- How to make Halo run on Xbox 360?
  - obvious way – add hardware, raise cost, ship in 2006
  - clever way – emulate Pentium III in software, ship in 2005
- Summer 2005 I joined Xbox to work on Pentium III emulation effort on PowerPC
- Hundreds of XBOX titles run on Xbox 360 flawlessly to where consumers don't even realize it is emulation
- Windows XP and iTunes on Xbox? **Project Helium** ☺

X86-on-PPC Technical Issues

- Emulating Pentium III on PowerPC *accurately* is extremely tricky technical problem:
  - X86 is little-endian, PowerPC is big-endian
  - X86 uses 80-bit floats, PowerPC 64-bit floats
  - Pentium III is out-of-order core that can retire 3 instructions per cycle, while the Xbox 360 (and PS/3) core is in-order dual-issue
  - Completely different device hardware
- But, had 4:1 clock speed advantage to work with
  - Ultimately achieved the >500 MIPS performance necessary, no extra hardware costs required!

A Sample Emulation Problem

- X86 performs arithmetic at 8-bit, 16-bit, and 32-bit sizes on Pentium III and produces 6 arithmetic flags: SF ZF CF OF AF PF
- PowerPC only performs 32-bit arithmetic and only has ZF CF OF and LT and GT flags
  - How to generate flags for 8-bit and 16-bit operations?
- An x86 compare instruction “CMP” performs both signed and unsigned compare at once
- PowerPC has unique signed and unsigned compare instructions, not one like x86
  - Don’t know ahead of time which x86 needs

Lazy Flags Evaluation

- This arithmetic flags problem posed a huge problem in terms of implementation and performance
- I spent many sleepless nights solving it
  - Drove to work at 2am the night that I did solve it
- Solution turned out to be simple:
  - A technique called Lazy Flags Evaluation
  - Proven in other simulators in the past
  - The idea is to evaluate a particular arithmetic flag only when needed based on results of operation

Lazy Flags Implementation

- Most lazy flags implementations record a long list of different data:
  - Input values (usually two or three)
  - The resulting value (e.g. the sum or difference)
  - The operation that was performed (an enum)
  - The size of the operation (e.g. 8, 16, 32, 64)
- When a flags is needed the arithmetic operation is replayed and the correct flag value recorded
- The problem – this is far too slow!

Even Better Solution

- I derived a technique which required only three values to be recorded after any arithmetic operation, such as the 32-bit sign-extended result
- Most common x86 branches are JZ and JNZ, so if you know the result you can derive ZF easily
- Even developed a bit twiddling trick for when signed and unsigned compares not equally fast and you have to emulate one with the other
  - Granted U.S. Patent #7,752,028 for that in 2010
- Bochs 2.5.1 today records only 2 lazy flags values

Another Gotcha – Sign Extension

- Need to sign extend an integer, e.g. take 8-bit unsigned and signed extend to 32 bits
  - Some CPUs load unsigned bytes faster than signed
  - i.e. prefer MOVZX over MOVSX, LBZ over LBA
- Typical C/C++ trick is to shift unsigned value left then shift right signed integer, e.g:
  - int x = ((int)(u << 24)) >> 24;
- This is not an very good portable line of code!
  - e.g. P4 had no barrel shifter, 6 + 6 = 12 cycles total!
  - e.g. PowerPC chips tend to have slow signed right shift

Sign Extension in 2 Cycles

- Observe:
  - if N is unsigned integer of value 0 or 1
  - 2's complement negation gives:
  - -N = NOT(N) + 1, therefore…
  - -0 = NOT(0) + 1 = (0 XOR (-1)) + 1 = 0 = sign_ext (0)
  - -1 = NOT(1) + 1 = (1 XOR (-1)) + 1 = -1 = sign_ext (1)
  - Can also be written -N = NOT(N-1) = ((N-1) XOR -1)
- XOR and SUB are 1 cycle on any machine
  - sign extend from bit b, just replace -1 with (-1 << b)
  - (-1 << b) is just a constant if b known at compile time
  - sign_ext(N,b) = (N-(1<<b)) XOR ((-1)<<b)
- 2-cycle sign extend from any constant bit position

Leaving Microsoft (again!)

- In early 2007 nothing at Microsoft particularly interested me. Xbox 360 was done, Nirvana was done, so I left again in May 2007
- Decided to focus on open source tools, make an open source release of Gemulator (9.0 released in late 2008 bringing Atari ST to Windows Vista and later Windows 7, woo!), and explore clever new simulation techniques for better debugging and tracing

A Blog is Born

- September 2007 started the NO EXECUTE blog on my emulators.com site:
  - http://www.emulators.com/nx_toc.htm
- Focus on x86 issues, dig into the historical design reasons software is buggy, why it's bloated, and looking at better x86 design decisions
- Concluded that AMD goofed with 64-bit x86 extensions, that flat memory model is a bad thing, and that everyone need better tools

The Search for Nirvana II

- Major limitation of Nirvana, PIN, DynamoRIO is they act like debuggers, only tracing user mode applications and not the whole system
- What is needed is a full-system simulator – a PC emulator with efficient tracing and instrumentation capabilities
- QEMU 0.9 and Bochs 2.3.5 existed in 2007, both were slow, and QEMU too inaccurate

Improving Bochs

- Late 2007 I started tinkering with optimization ideas in Bochs
- Over the course of 6 months, Bochs performance improved 3x, from ~30 MIPS to ~100 MIPS (on a Core 2 @ 2.66)
- In other words, simulated Windows XP now booted in 90 seconds instead of 5 minutes before… that's progress
- #1 performance hit – branch mispredictions
  - A mispredicted branch costs 20 to 25 cycles, or the equivalent of up to 75 arithmetic operations – **death by too many "if" statements**
  - Bochs was typically taking 4 branch mispredictions per x86 instruction simulated when typically should take just one – the dispatch to the x86 instruction handler
  - If instruction handlers can do their work in under 25 cycles, an out-of-order core hides that cost, therefore 2.66 GHz / 25 cycles = ~100 MIPS.
  - Bochs was using inline x86 assembly to do arithmetic flags – I replaced that code with faster lazy flags code written in C++

From Bochs to Intel

- In May 2008, I co-wrote a paper with Bochs maintainer **Stanislav Shwartsman** to describe how Bochs and Gemulator were optimized to each reach 100 MIPS emulation speeds.
- Paper was accepted for ISCA workshop AMAS-BT and we presented the paper in Beijing
  - http://www.emulators.com/docs/VirtNoJit_Paper.pdf
- A few months later, Intel made me a job offer to come work on simulators for them ☺

Microcode Simulation

- At Intel I worked on simulating prototype microcode of future CPU cores (way cool!)
- Vital in designing CPU architectures because a modern out-of-order pipeline executes micro-ops (the output of decoded x86 instructions)
- Such a simulator can be used to model wider pipelines, new micro-ops, and tweak various parameters without actually taping out real silicon (which is costly and time consuming)

From 10 MIPS to 200 MIPS

- Microcode simulation is typically even slower than x86 simulation:
  - A single x86 instruction can decode into multiple micro-ops, so more of them to simulate
  - CPU retires multiple micro-ops per clock cycle, so atomicity across multiple micro-ops is required.
  - i.e. if a LOAD, an ADD, and a BRANCH all retire at the same time but the LOAD faults, the side effects of the ADD and BRANCH have to be rolled back
- The existing simulator ran at all of 10 MIPS
  - Booting Windows in simulation could take hours

Make that ***300*** MIPS

- During 2009 and part of 2010, my former Xbox colleague **Jens Troeger** and I implemented a fresh microcode simulator from scratch using our tried and techniques from Bochs, Gemulator, Helium, and Nirvana.
- Resulting simulator peaks at ~~200~~ 300 MIPS ☺
- Our paper describing this work published at ISCA 2011 in San Jose this past June:
  - http://amas-bt.cs.virginia.edu/2011proceedings/amasbt2011-p3.pdf

Amazon EC2

- In summer of 2010 I left Intel and joined Amazon here in Seattle to work on the EC2 infrastructure, particularly to roll out Windows server support on the Cluster Compute cc1.4xlarge instances

- Job involved everything from Linux kernel and hypervisor, to understanding hard disk and SSD characteristics, networking, VM scheduling, load balancing, Remote Desktop, customer and vendor engagement, virtual clock drift, and lots and lots of benchmarking

- Shipped the working product September 2011
  - http://aws.typepad.com/aws/2011/09/now-available-windows-server-2008-r2-on-cluster-compute-and-cluster-gpu.html

- Not using EC2? Duuuuude! http://aws.amazon.com/

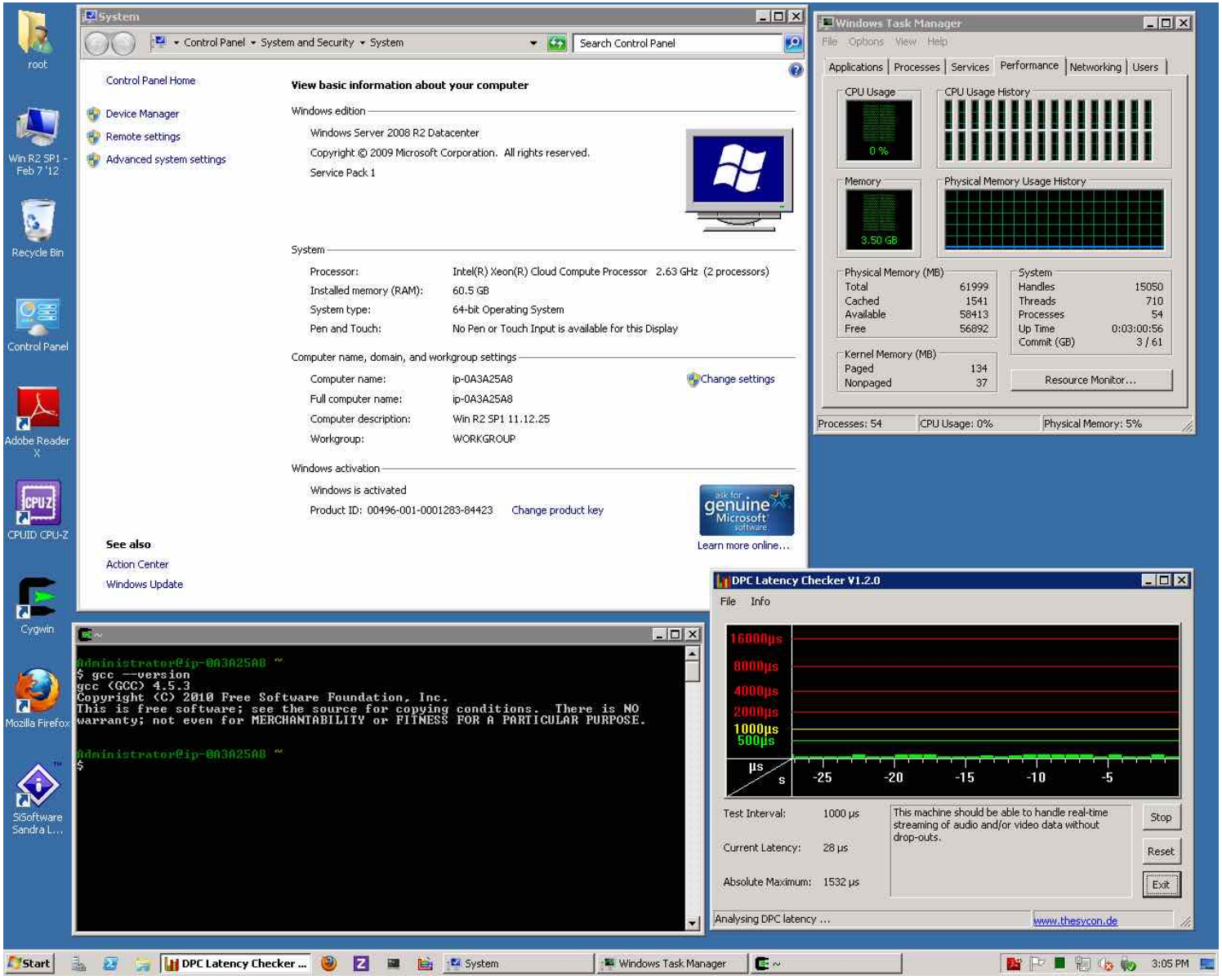
Team Amazon EC2 on Prime95

- Needed to stress the Windows VM stack
  - What better way to stress a machine than running PRIME95 on every core!
- We quickly made Mersenne top producers:
  - http://mersenne.org/report_top_teams/
- 168685 GHz-Days and counting = 4 PHz-Hours
  - Equal to 6 months of 100 quad-core machines ☺

cc2.8xlarge

- Amazon's largest instance type today
  - http://workspresso.wordpress.com/2011/11/17/amazon-ec2-beefs-up-hpc-cloud-with-cc2/
  - Rates at 88 ECU, equal to about 44 Core 2 cores
  - Based on 16C/32T @2.6 GHz Xeon processors
  - Delivers over **200 GFLOPS** per instance
  - In November 2011 a cluster of cc2.8xlarge instances produced the world's 42nd fastest supercomputer in the Top 500 list: http://top500.org/list/2011/11/100

60GB, 32 threads, $3/hr. Sweet!



Seattle's *new* Evil Empire? ☺

- Amazon's campus downtown getting bigger… and bigger
- With a Starbucks in every lobby, hooray!
- Downtown location definitely better than farmland and forest

2012 and Beyond

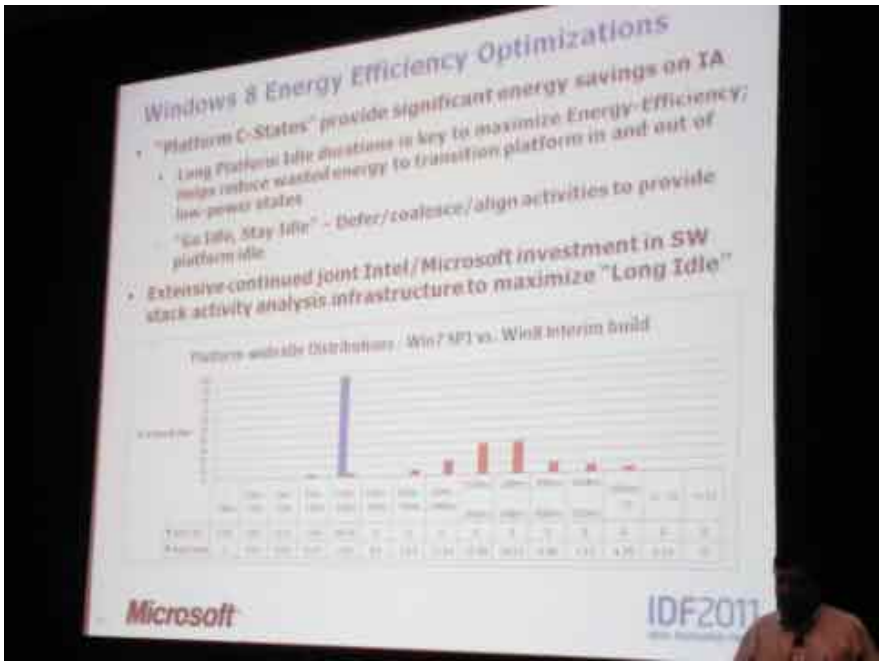
- My end of year 2011 blog posting looked at a number of interesting technologies here or on the way in 2012/2013:
  - Windows 8 on ARM and tickless kernel
  - Ultrabooks and tablets
  - Intel Sandy Bridge, Ivy Bridge, and Haswell
  - AMD Bulldozer and new direction of Core Next
  - Improvements in solid state and hybrid drives
  - AVX and other micro-architecture improvements
  - http://www.emulators.com/docs/nx34_2011_avx.htm

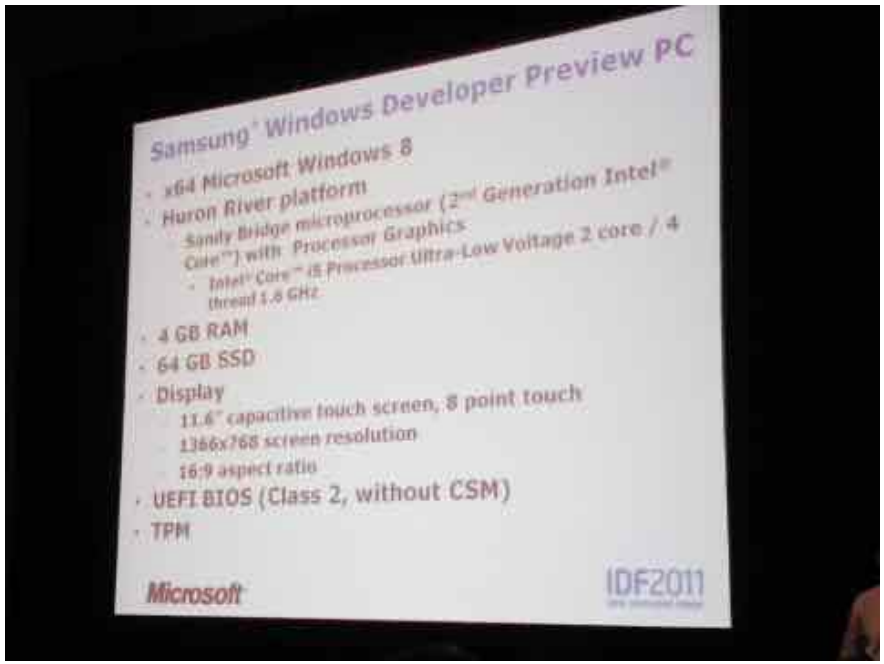
Slim Is In

- Win7 “Ultrabooks” and tablets are here today
- Based on latest Sandy Bridge processors
- 4GB RAM, 128GB SSD, 1366x768 standard

Windows 8

- First Windows to run on x86, x64, and ARM
  - http://blogs.msdn.com/b/b8/archive/2012/02/09/building-windows-for-the-arm-processor-architecture.aspx
- First “tickless” Windows kernel (finally!)
  - Great for virtualization and power saving since it reduces how often an idle processor needs to wake up





Why Sandy Bridge Rocks

- Not just an incremental Core i7 improvement
- Introduces 256-bit AVX extensions to SSE
- AVX supports 3-operand operations and replaces klunky REX prefix with VEX prefix resulting in denser instructions – 11 byte max
- Integer arithmetic improvements deliver 15% to 25% speedup in IPC over previous Core i7
- Fewer partial EFLAGS stalls and fast PUSHF – great for emulation and binary translation

Ivy Bridge

- Is the 22nm die shrink of 32nm Sandy Bridge
- Will have ½ the power consumption of Sandy Bridge for same level of performance
- Tablets and Ultrabooks should have practically all day battery life
- RDRAND instruction adds true random number generation
- REP MOVSB finally does what you expect

Haswell

- 22nm new “tock” micro-architecture in 2013
- Adds AVX2 (integer extensions to SSE in the same way AVX extended floating point SSE)
- 3-operand integer, hmmm, RISC like code in AVX, effectively doubles register file size
- BMI (Bit Manipulation Instructions) finally add efficient bitfield insert and extract to x86!
- Will completely change how interpreters and binary translation can be implement, ***can’t wait!***
- Read up on **butterfly shifters**, way cool stuff:
- http://palms.princeton.edu/system/files/IEEE_TC09_NewBasisForShifters.pdf

Why Butterfly Shifters Are Cool

- Traditional barrel shifter has N^2 complexity as number of bits increases
  - Each bit has fan-in of N and fan-out of N. Ideally you want gates to have fan-out 4 (FO4), not FO64!!
- Log shifter is a multi-stage shifter where each stage shifts some power of 2 bits – 16, 8, 4, 2, 1
  - Complexity drops to Nlog2N, fan-in fan-out = 2
  - Each stage is pipelined, can have multiple shifts in flight
- Butterfly shifter allows permuting bits making possible shifts but also bit reversal, byte swapping, byte broadcast, extension from arbitrary bit, bitfield extract, and parallel extract and deposit (PEXT/PDEP)

Solid State Drives

- Currently about $2/GB and dropping
- A plug-in PCIe based SSD delivers **>1300MB/s**
  - I personally use the OCZ RevoDrive products
- SSD is standard in thin form factors
- Hybrid drives really work, great price/perf
  - I personally use the Seagate Momentus XTM
  - Mechanical disk drive with SSD cache
- 10000 IOPS is the new 100 IOPS
- <1 ms access latency

AMD's New Direction

- June 2011 here in Seattle AMD announced "Core Next", a new 64-bit GPU architecture with calls, exceptions, cache coherency, page tables, *hmmm, sounds like new general purpose CPU core too?*
- In my blog I suggest AMD is moving away from x86, possible to use DBT to emulate x86
- February 2012 AMD says "not married to x86"
- We'll see, but this slide deck ***excites me a lot***:
- http://developer.amd.com/afds/assets/presentations/2620_final.pdf

Exciting Next Few Years

- Ivy Bridge brings 256-bit computing everywhere – start coding in AVX today
- Haswell will improve emulation performance with BMI
- AMD may be developing versatile ISA-agnostic core – emulate anything using a “GPU” core?
- Bochs 2.5.1 is now emulating AVX, AVX2, BMI at 100 to 200 MIPS speeds – use it!
  - http://bochs.sourceforge.net/
- The cloud costs $1 to $3/hr to test your code on latest high core count hardware, cheap!
- Apps are the new wave of consumer computing

Apps are the new Commodore 64

- In the 1980's, the Apple II, Atari 800, C-64, VIC-20, TRS-80 and other home computers were intended to bring computing into the homes of millions – if you like to program in BASIC
- So easy a soccer mom could use
  - "Keep recipes in a computer" paradigm… FAILED!!
  - But teenage boys like me did learn to program
- iPad, Android, WM7 ***are*** the new wave of home computing that soccer moms really ***do*** use
- OSes, SDKs are free – Linux, gcc, even Windows 8 and Visual Studio betas are free downloads

VM Opportunities Abound

- In 2007 I predicted next killer app would be running Windows machines on the web, see:
  - http://www.emulators.com/docs/nx07_vm101.htm
- In 2011, I helped make that reality
  - **Don't just complain, get off your butt and fix things!**
- However, not the full solution I dreamed of:
  - No migration to/from the cloud to local device
  - No way to emulate future hardware or hide existing hardware or fake a specific CPU and ISA for legacy apps
  - No nested virtualization support
  - No trace/playback/TTD options for developers

Beef: Why I Hate Hate Hate VT-x

- You can't run Hyper-V or VMware recursively using hardware virtualization
  - Because most VMs leave code in the same ring of execution
  - user mode code in Ring 3, kernel code in Ring 0
- X86 has always supported 4 levels of privilege
- "Ring compression" is a technique used by Virtual PC and older VMware which moves guest Ring 0 to host Ring 1 – that doesn't nest
- When AMD speced 64-bit mode **they removed key features** necessary for ring compression and binary translation techniques, e.g. segment checks
- AMD and Intel **had to invent VT-x**, as some incorrectly call it "Ring -1"
- VMware, Microsoft, Xen and others blindly switched to using VT-x
  - Doesn't nest! You have to emulate VT-x anyway to have hope of nesting
  - Recent paper on nesting VT-x using simulation suggest "VMEXIT explosion"
- VMware engineers published a paper in 2006 raising concerns about VT-x
- VMware paper in 2009 suggesting DBT in 64-bit long mode can be faster
- If interpreter and translated code always ran in Ring 0, would be nestable

Consider This…

- Merge the separate concepts of VM hypervisors (Hyper-V, Xen, VMware, VirtualBox) and simulators (Bochs, QEMU) and debuggers (Nirvana, PIN, DynamoRIO) in a single unified VM engine

- Requires DBT, interpretation, and dynamic instrumentation to give a sliding tradeoff between performance and sandbox granularity

- I believe that an untapped market is for "**faster-than-bare metal**" legacy virtualization. i.e. take single-core 32-bit VMs running DOS/Win98/WinXP and translate to 64-bit long mode to take full advantage of 64-bit and AVX. Sort of like "Windows XP Mode" in Windows 7, but done right and with TTD capabilities

- This is the kind of opportunity a small startup could tackle, cheap. Haswell and Core Next show promise for DBT and nested VMs.

Hot Off The Press!!!

- As I was giving this presentation on February 7, Intel publically disclosed new transactional memory technology for Haswell called **HLE** (Hardware Lock Elision) and **RTM** (Restricted Transactional Memory)
- http://software.intel.com/en-us/blogs/2012/02/07/transactional-synchronization-in-haswell/

- HLE are prefixes placed in front of CMPXCHG, LOCK ADD, and other atomic operations to make interlocked operations faster in legacy code – just use it
- RTM adds new instructions to allow for a limited amount of memory to be updated as atomic transaction – think **multi-word compare-swap** on x86

Haswell Changes Everything

- HLE and RTM present research and product opportunities
- Think about how you would rewrite these using transactional memory: a thread-safe lock-free linked list, a lock-free hash table, a multi-threaded Java VM…
  - ***Could a Java VM using RTM outperform legacy native??***
- HLE will reduce overhead of synchronization primitives and thus improve legacy lock performance and scaling
- RTM eliminates the need for most locks and critical sections by allowing atomic update of entire data structures even in shared memory
  - Other threads do not see a partial update with incomplete data
- **Bochs 2.5.1 needs HLE/RTM simulation to be added**
  - **Join the effort and contribute! http://bochs.sourceforge.net/**

Computing in the 21st Century

- Chuck Thacker, Turing Award Winner, gave a keynote at 2010 ISCA conference suggesting we reconsider how computing for 21st century:
  - Why still have demand paged memory? RAM is plentiful
  - Why have so many exceptions and exception handlers?
  - Do we really need pre-emptive multitasking? Reconsider task scheduling
  - Programmers can't get threading right, are locks and shared memory the answer?
- http://www.microsoft.com/presspass/features/2010/mar10/03-09chuckthacker.mspx

- Others suggested complexity of hardware and software is in deadlock
- Intel has a "**H**ardware/**S**oftware **C**o-**D**esign" group that seeks to bridge the chasm between hardware knowledge and software knowledge
  - The two disciplines cannot exist in isolation
  - Otherwise leads to more of mess of past 20 years
- All sorts of opportunity for new CPU, OS, and language designs
- That goes well with the idea of virtualize everything, simplify hardware
- And most importantly – **simulate everything**, don't just assume
- CGO **WISH** workshop and ISCA **AMAS-BT** workshops focus on HSCD

Rethinking Algorithms

- As we rethink hardware design we need to also rethink decades old computing algorithms
- That was the subject of my 2008 talk at Conestoga “Parallelism: The Real Y2K Crisis”:
  - http://www.emulators.com/docs/TheRealCrisis.pdf
- Dr. Dobbs just stressed it again this week:
  - http://drdobbs.com/parallel/232500147
- Data structures need to support both “big data” as well as high levels of parallelism on today’s highly multi-core processor and cluster machines

Counting Lots of Things

- Consider the case of a large data set and needing to know if some symbol S is in that set.
  - S could be a 10-digit phone number, a word, etc.
- If the set is millions of objects (words, phone numbers) you can use a bit vector or array of booleans, easy!
  - Memory footprint bounded in the low megabytes
- Even a set of all possible 32-bit addresses doable
- If you need to not just know if a symbol is present but also how many times, you need counters
- Bloom Filter can handle “is” but not “how many”

The Count-Min Sketch Algorithm

- To count "how many" in large sparse dataset, might use hash table
  - Large memory footprint and lots of bucket collisions
- Or use cheap RAM and large pagefiles on hard disk
  - I've written profiling code in Bochs to count accesses of all 32-bit addresses using a 2^32-entry array of 64-bit counters – 32GB of footprint
- But what if I'm tracking counts of addresses in a 64-bit process, which has 2^43 = 8TB of user mode address space?!?
  - 8 trillion possible symbols with 64-bit counters needs 64TB of array, yikes!
- Count-Min Sketch is newer algorithm that uses multiple hash tables, each with unique hash function, to keep a "close enough" track of counts:
  - http://dimacs.rutgers.edu/~graham/pubs/papers/cmencyc.pdf
  - If you collide, don't worry, just update the counter anyway
  - When querying the count of a symbol, look up all the hash tables and choose the *lowest* count, i.e. the one that had fewest bucket collisions
- "Big Data" requires thinking up new ways to process terabytes of data where fuzziness is ok – e.g. return "thousands" search hits, not "3485"

Looking for a Job

- I've interviewed dozens (hundreds?) of candidates while at Microsoft, Intel, Amazon - most of them I rated "No Hire"
- My #1 beef – padding the resume - **DON'T DO IT!**
  - If you claim to know x86 assembly but don't, I will nail you to the cross in a matter of minutes and eat your brain
  - Saying you're "willing to learn on the job" is not an answer if you claim to already know something
- Just be honest and list what you know and enjoy
  - Write papers, blogs, apps - **create experience, show interest**
- Target a group/department that you want to interview with at a company and use LinkedIn to network with potential employers directly – get in their face to get noticed
- **Virtualization a hot job category these days, just saying ☺**

Never Easier to be an Entrepreneur

- 1980's – thousand of dollars of equipment costs, slow internet, months to develop something and get it published, disk duplication, distribution, trade show hassles, Mac / PC / Unix not interoperable

- Today - $700 desktop or $900 laptop with latest Sandy Bridge technology is all you need, free Linux/Windows development tools, self-publish your apps to web, get famous, get paid, get hired

- Learn Linux, gcc, Java, C#, Win8, Android – doesn't cost anything, doesn't require a degree, no excuse to put off learning

- **No whining!** ***You*** will be as successful as how much effort you put in and risks you are willing to take. I made big gambles in my life and worked 80 hour weeks for years – don't expect easy shortcuts.

Further Reading and Viewing

- For detailed postings and product reviews related to the topics discussed, read my **NO EXECUTE** blog at: http://www.emulators.com/

- ***Do*** shamelessly promote yourself and your hard work, as I did in these 1992 Gemulator demo videos to spread the word about Atari ST emulation (bad hair, cheap production, and all):
  - http://www.youtube.com/watch?v=UNOw3eyBygw
  - http://www.youtube.com/watch?v=RxZOAc5UI1w
  - http://www.youtube.com/watch?v=tuG8l6epKug

Thank You!

- I'd like to thank the Conestoga College students who came out to my talk on February 7th and who heroically held their bladders while I went on and on and on and on for almost 4 hours before Ignac kicked me out.

- Questions? darekm@emulators.com