

What is Digital FX!32?

- ✍ An end-user utility for Fast transparent eXecution of Win32 applications on Alpha NT
 - ñ Transparency:
 - ï Apps install & launch ìjust like x86î
 - ñ Correctness:
 - ï Apps ìjust workî
 - ñ Performance:
 - ï Target is 70% of native Alpha
 - ï On 500 MHZ Alpha: approx. P6 (200)

Step 1. User executes app...

- ✍ NT invokes FX!32 Runtime via transparency agent
- ✍ Runtime:
 - ñ loads the x86 .exe and associated x86 .dlls
 - ñ queries Database for translated images
- ✍ Seeing none, Runtime uses Emulator to run the application, producing:
 - ñ correct application behavior
 - ñ execution profiles

Step 2. In the background...

- ✍ FX!32 Server:
 - ñ sees new profile data
 - ñ evaluates whether to optimize
 - ñ invokes Translator
- ✍ Translator processes application, producing:
 - ñ translated image in a native Alpha .dll
- ✍ FX!32 Server manages execution profiles and translated images:
 - ñ merging, purging, etc.

Step 3. User executes app again

- ✍ NT invokes FX!32 Runtime via transparency agent
- ✍ The Runtime:
 - ñ loads the x86 .exe and associated x86 .dlls
 - ñ queries Database for translated images
- ✍ Seeing some, Runtime:
 - ñ loads translated .dlls
 - ñ uses optimized Alpha code when available
 - ñ uses Emulator otherwise

New Features

- ✍ A new style of user interface (No commands)
- ✍ A long term profile data base (Per user per program kept automatically and out of sight)
- ✍ A translator that converts important parts of a program to native code (in the background)
- ✍ Its not a JIT
- ✍ Its a JAC (Just After Compiling)

Transparency Agent

- ✍ Provides transparent execution of 86 applications
- ✍ No user commands required
- ✍ Present in all Alpha processes
- ✍ Invokes FX!32 Runtime to execute x86 images
- ✍ Acts like an extension to the OS, but it's user level code

FX!32 Agent

- ✍ Enables processes - starting with login shells and servers
- ✍ Loops through all the images in the process into which it is loaded and hooks APIs.

- ïCreateProcess
 - ïLoadLibrary
 - ïFreeLibrary
 - ïLoadModule
 - ïWinExec
 - ïGetProcAddress
 - ïGetPrivateProfileString

FX!32 Agent

CreateProcess

- ñ Determines if x86 image by looking at image header
- ñ If not x86 image, creates the process and injects self if image is `injectable`
- ñ If x86 image, starts FX!32 to run image

FX!32 Agent

GetProcAddress

ñ Hooks APIs accessed by GetProcAddress

FreeLibrary/LoadLibrary

ñ For x86 images calls FX!32 FreeLibrary

GetPrivateProfileStringA

ñ Returns original shell when called with parameters
`ìBOOTî, `ìShellî.

Emulator

- ✍ Interprets x86 code
- ✍ Allows calls to translated routines and jackets
- ✍ Collects profile data
- ✍ Written in Alpha assembler - Icache aware
- ✍ x86 state maintained in Alpha registers
- ✍ Caches used for profiling
- ✍ x86 registers (also in Alpha registers)
- ✍ Condition code state (also in registers)
- ✍ FP status and control words

Run Time Jackets

- ✍ If a native version of a library routine exists, a jacket routine is used to convert arguments and make the native call
- ✍ FX!32 jackets 40 DLLS, 12,000 WIN32 call APIs
- ✍ An application that makes a lot of GUI calls can spend half of its time in native code
- ✍ Most jackets are generated automatically from header files. Some are hand built, some are generated on the fly

Generation of Profiles

- ✍ Profiles contain information about addresses
 - ñ Call targets
 - ñ Targets of indirect control transfers
 - ñ unaligned memory accesses

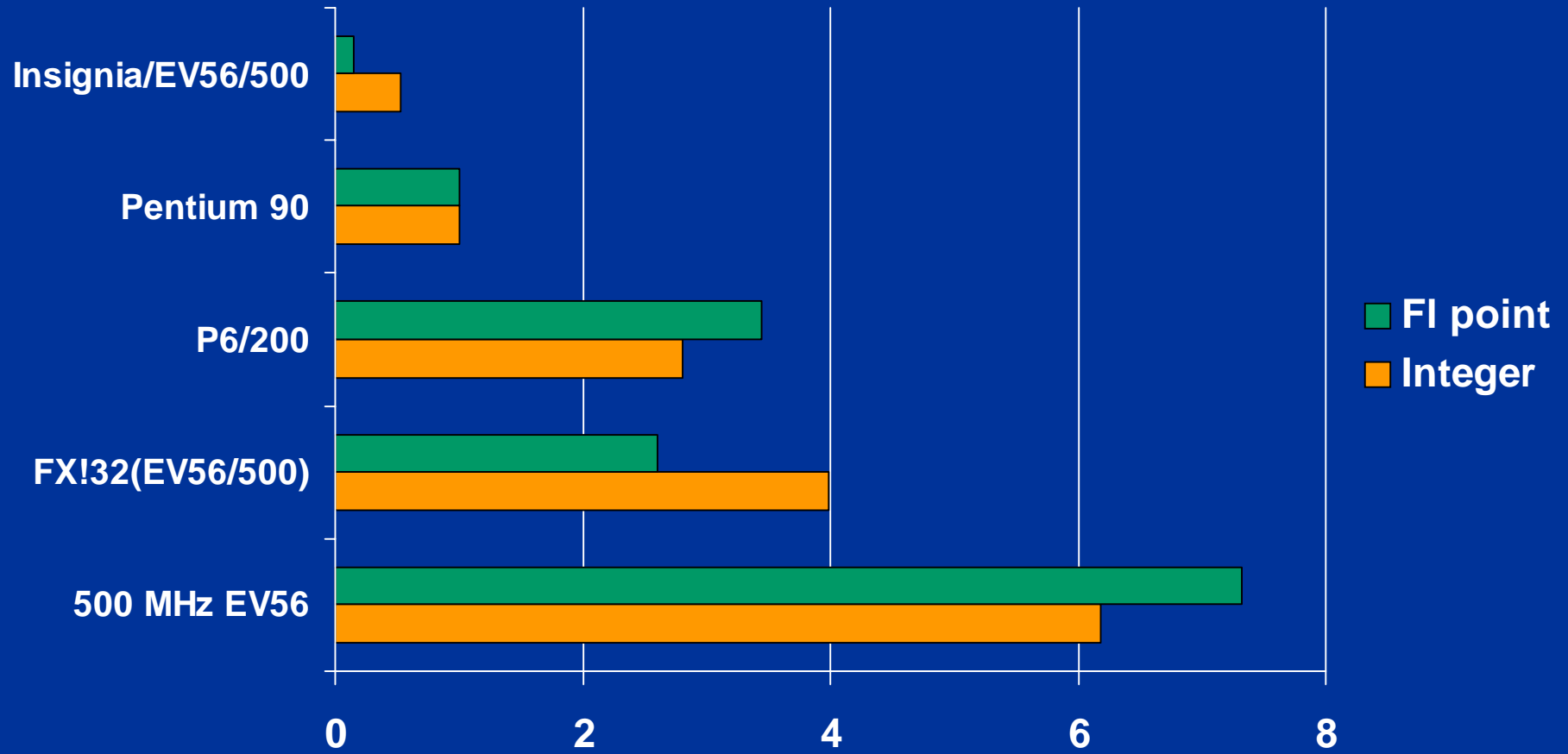
FX!32 Server

- ✍ Implemented as a regular NT service
- ✍ Merges run profiles into cumulative profile
- ✍ Runs translator when needed
- ✍ Maintains profiles & translated images
- ✍ Can share translation results

FX!32 Translator

- ✍ Need to find routines (based on profile)
- ✍ Only translates routines that were executed
- ✍ Translates most heavily executed code first
- ✍ No type information
- ✍ Uses profile data to determine alignment
- ✍ Cannot count on any call argument rules (or much else)
e.g., programs can reach up into parent's stack and change something

BYTE Benchmark



What's Running Today

4nt
Adobe Photoshop 3.05
Arm2
Boggle
Borland Paradox 7.0
CodeWright 4.0
CorelDraw 6.0
CRT 1.1
Dispcalc
Drag and file 2.0
Drag and View 1.22c
Drag and zip 1.14a
Fury 3.0
Hangman
Harvard Graphics 4.0
KAI Power Tools
Kloitz
Live Markup
Lotus Notes 4.0 client
Mathcad 6.0
Mazelord
MicroEmacs
MicroGrafix ABC Flowcharter 6.0
MicroGrafix ABC Graphics 6.0
MicroGrafix Design

MicroRim Rbase 5.1
Microsoft Access 7.0
Microsoft Excel 7.0
Microsoft FrontPage 1.0a
Microsoft Office95 Pro
Microsoft Powerpoint 7.0
Microsoft Publisher 3.0
Microsoft Schedule95 7.0
Microsoft Visual Basic 4.0
Microsoft Word 6.0
Microsoft Word 7.0
Netscape Navigator 2.01
O'Reilly Website
Persuasion
Pixfol32
Polyview
Solitaire
Stonehenge
TinyCAD
Visio Technical 4.0
Visual Reality 2.0
Visual Slick Edit
WinVN 0.99.7
WinZip 6.1
Adobe Acrobat

Corel Xara
CU-SeeMe
DacEasy Accounting Payroll
DeltaGraph Pro 3.5
Frame FrameMaker
LiveView Pro
Macromedia Freehand
Microsoft Money95
Microsoft Works
Mosaic V2
Novell WordPerfect 6.1
Street Atlas USA 3.0
Visual FoxPro
Adobe PageMaker 6.0
Microsoft Encarta Encyclopedia
Microsoft Golf95
Ray Dream Designer 4.0
RoboHelp95 3.7
Tcl/Tk
ThumbsPlus32
Trispectives
Visual Software Instant 3D

...and many more

Digital FX!32: Can't Do It All

- ✍ Win16 unsupported
- ✍ Drivers unsupported
- ✍ Debug API unsupported

Licensing Issues

- ✍ None with Digital FX!32!
- ✍ **CANNOT** run application on Alpha without original x86 code
 - ñ Digital FX!32 enforces same software license as the original application
 - ñ Translated code can be moved around freely, but useless without original application
- ✍ Vendors can rest easy that their intellectual property is safe