Pierre Paulin Biography

Born in Canada, dual citizenship (France)

Education

- Engineering Physics  B.Sc.  U. Laval  1982
- Electronics  Ph.D.  U. Carleton/BNR  1988

Experience

- Renault, Paris  Robotics  1982
- BNR/Nortel  H/W synthesis, C compilation  1984 - 1994
- STMicro.  Embedded Systems  1994 - now
Towards Application-Specific Architecture Platforms: Embedded Systems Design Automation Technologies

Pierre Paulin
Director, Embedded Systems Technology
Central R&D

STMicroelectronics
STMicroelectronics

Franco-Italian origin (formerly SGS-Thomson Microelectronics)

International company now, but with strong roots in Europe
STMicroelectronics POSITIVES

- French cuisine
- Driving german cars on german highways
- Italian passion
- Swiss sense of organization
- English courtesy
- Canadian real estate
- U.S. stock options
STMicroelectronics NEGATIVES

- U.S. stock options (one week later)
- Canadian weather
- French courtesy
- Driving french cars on London motorways
- English food
- Swiss passion
- Italian sense of organization
Introduction

Part I: System design platforms
- Sample design platform: ST set-top box
- ASAP: Application-Specific Architecture Platform
- Which platform components?
  - H/W vs FPGA vs AS-Processors vs GP-Processors
- Platform design automation challenges

Part II: ST platform automation technologies
- ST system design environment
- ST embedded software development tools
Bridging the Process / Design Gap

- **Process capability** +50% / year
- **RTL design capability** +20% / year

**Sources:**
- System Level Entry
- HW-SW Methodologies
- Platforms, IP Reuse
- RTL-to-layout Flow

**Design complexity** (transistor count)


Source: ST
System Design Technology Strategy

Complement ST's rich, heterogeneous process technology offer with tools/methods supporting it at architecture and system design levels

- Domain-specific tools: DSP, control/protocol, real-time
  - Executable specifications, system validation
  - DO THE RIGHT THING

- Processors, memories, logic, datapaths, analog, RF, A/D
  - Virtual prototype, Architecture exploration, codesign
  - Platform-based design
  - High performance S/W compilation
  - DO THE THING RIGHT, FAST

- Logic, analog, DRAM, SRAM, Flash, High-speed, Power
  - DO IT IN ON A SINGLE CHIP!
Part I:
System Design Platforms
ST Platform Example: Set-top box, DVD

Development of complete IP foundation and design platform
✓ Front-end, demux
✓ Std/High-def MPEG2 video/audio, Display, 2D/3D graphics
✓ RISC, AS-DSP, smart card, I/O

Led to ST world leadership position
● MPEG2 audio/video decoder -> 30 million chips sold
● Digital satellite set-top box chips -> Over 70% market share
Central R&D

Platform Approach = Fast Derivatives

ST Digital TV Product Families

Basic Pay TV
DISH, DIRECTV

DVB Interactive
BSkyB, Canal+, ONdigital, TPS
MediaOne, US Cable DCT2K, Explorer 2K

Web Interactive
MediaWeb, DISH Player,
DVB-MHP, DIRECTV++, UK Cable

STi7000 family
HD decoding, HD display,
3D graphics, High quality display

OMEGA2
2D/3D graphics, Hard-Disk Drive,
Multiple demux, Java, WinCE

OMEGA1
Still image plane,
BLT engine, Enhanced display

OMEGA
Integrated Audio/Video, Demux
Micro, On-screen Display

HD/3D Capable
OpenCable
Application-Specific Architecture Platform (ASAP)

Reconfigurable Platform Template

- Config DSP, MCU
- uProg I/O proc.
- MMedia processor
- Standard Processors

Derivative

Mapping

Executable system specification

- Domain-specific languages
- System-oriented C/C++ environments

Programmable Platform Instance

- Fixed Silicon with high programmability:
  - S/W on general purpose and domain-specific processors
  - Microprogrammable I/O processing
  - FPGA coprocessors

Central R&D
Platform Example: Set-top Box

- Analysis of application requirements
- Derivation to specific platform instance

Central R&D
Platforms: Divide and Conquer

**Application IP**
- GSM, UMTS
- Set-top box, DVD, HDTV
- xDSL modem, Internet switch
- Car multimedia
- SmartCard
- Hard-Disk Drive

**Platform "Insulation"**

**Platform IP**
- Wireless terminals
- Multimedia, Graphics

**Microarchitecture "Insulation"**

**Component IP**
- Commodity cores: DSP, MCU, RISC, Bus
- A/D, D/A, I/O, Analog
- Flat Panels
- Libraries: Cells, memories, I/O

- System-level expertise
- S/W expert
- Platform programming

- Domain-specific platform expertise
- Fast platform derivatives

- Component expertise
- Process differentiators (BICMOS, RFCMOS, eDRAM, eFlash, OTP)

- System function

- H/W-S/W architecture

- RTL
- Layout

- MPEG2 decoder
- RAM
- 2D graphics
- ST40
Platform Architecture Definition

- Requires combination of four competences

- Short Time-To-Market

- Long Platform Lifetime

- Amortize increasing NRE costs
e.g. mask set 250 -> 500 K$
Which Platform Components?

**Application IP**
- GSM, UMTS
- Set-top box, DVD, HDTV
- xDSL modem, Internet switch
- Car multimedia
- SmartCard
- Hard-Disk Drive

**Platform "Insulation"**

**Platform IP**
- Wireless terminals
- Multimedia, Graphics

**Microarchitecture "Insulation"**

**Component IP**
- Commodity cores: DSP, MCU, RISC, Bus
- A/D, D/A, I/O, Analog
- Flat Panels
- Libraries: Cells, memories, I/O
## Design Platform Component Options

### High design cost
- Low flexibility
- Low power/operation
- High speed
- Lowest part cost

### Low design cost
- High flexibility
- High power/operation
- Lower speed
- Medium part cost

<table>
<thead>
<tr>
<th>Component</th>
<th>Analog H/W</th>
<th>Structured Custom</th>
<th>RTL Flow</th>
<th>FPGA</th>
<th>FPGA &amp; Processor</th>
<th>Config. Processor</th>
<th>DSP MCU</th>
<th>GPP (General Purpose Processor)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Design cost</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Flexibility</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Power/Op</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Speed</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
</tbody>
</table>
Processor Evolution Case Study

- 1992: MPEG1 ASIC
- 1995: MPEG2 ASIP
- 2000: Audio DSP

Processors:
- Structured Custom
- RTL Flow
- FPGA
- Config. Processor
- DSP
- MCU
- GPP (General Purpose Processor)
Component Usage Trends

1990’s
- Structured Custom
- RTL Flow
- FPGA
- FPGA & Processor
- Config. Processor
- DSP MCU

2000’s
- ASIP
- GPP (General Purpose Processor)
Embedded Processor Market Share

1997

13.5 B$ (4.2B parts)

- 8 bit MCUs: 46%
- 16 bit MCUs: 20%
- 16-24 bit DSP (int): 21%
- 32 bit MPU/MCU: 10%

2002

30.8 B$ (9.1B parts)

- 8 bit MCUs: 31%
- 16 bit MCUs: 19%
- 16-24 bit DSP (int): 29%
- 32 bit DSP (flt): 4%
- 32 bit MPU/MCU: 17%

Source: InStat 1998
Dataquest 1998

Average price < $3

8-16 bit MCU + 16-24 bit DSP
= 80% 2002 revenues
95% 2002 volume

Average price ~ $15
Competitive Differentiation

% of Product Functionality

Performance/Cost/Power Differentiation

Key Differentiators

- IP reuse
- Executable specification
- Low-power/high-speed design
- Rapid prototyping

Application Algorithms

Processor Architecture  Tools for S/W productivity

System application knowledge

Central R&D
Sony PlayStation 2

- H/W
- ASIP
- DSP
- RISC

- I/O: FireWire, USB, TV RGB
- Physical simulation VLIW
- Geometry calculation VLIW
- MPEG2 decoder ASIP
- Image rendering ASIP
- Shading ASIP
- I/O processor
- One Standard RISC (MIPS III)
- Audio DSP
- [Modem DSP]

- Fixed design for multiple technology generations
- Little or no profits on hardware in short-term
- Significant revenue proportion from applications
- New product variations in S/W and F/W only
Network Processor

- SRAM, SDRAM
- CAM
- Policy engines

- Forwarding Engines (typically 6 to 8)

- One Standard RISC

- Programmability, flexibility
  V.S. cost/performance

- Application of Bermuda triangle!

- Application Algorithms

- Bermuda

- Processor Architecture

- Tools for S/W productivity

- System application knowledge
Heterogeneous multi-processor platform:
- Application specific processors:
  - Video DSP, Network proc., I/O
  - Domain-specific config. DSP: audio, low-power wireless
  - Domain-specific config. MCU: protocol processing, bit manip.
  - General purpose RISC, VLIW

Decreasing Risk
Increasing differentiation
Key Platform Automation Technologies

Reconfigurable Platform Template

- Config DSP, MCU
- uProg I/O proc.
- MMedia processor
- Standard Processors

Platform derivative generation

- Estimation
- Arch. configuration
- H/W synthesis
- Physical design

Executable system specification

Programmable Platform Instance

- IEEE 1394
- ST40
- MPEG2 decoder
- S/W

Application to platform mapping

- Allocation FPGA-S/W
- S/W compilation
- FPGA synthesis
- Interconnect configuration
Key Platform Automation Technologies

- Domain-specific languages
- System-oriented C/C++ environments

- Partitioning, allocation to multi-processors
- Efficient use of memory & communication
- High-performance S/W compilation for domain-specific processors
- Multi-processor RTOS
Platforms make commercial sense
✓ ST leadership in set-top boxes, MPEG2 chips

Decrease Design NRE
Time-to-market

Vision for next decade
- Embedded Software on heterogeneous multi-processors
  ✓ Application specific processors
  ✓ Configurable, domain-specific processors
  ✓ General purpose RISC, VLIW

Decrease: Fabrication NRE
Time-to-volume
Part II:
ST Platform Design Automation Technologies
ST System-Level Design Flow

System function
Matlab  SystemC
SystemStudio

System S/W Architecture
Appln. Stacks
Device Drivers

Evaluation/Partitioning
CoWare
FlexPerf: Performance eval.

System H/W Architecture
SystemC (Synopsys, CoWare)
FlexSim2: Arch. modelling

S/W design
Fixed-point DSP S/W refinement
FlexCC  GH/PGI, BSO, SH
FlexGdb  Multi, Gdb, Inquest
FlexSim  CHESS, C
FlexPerf s/w analysis

Interface design
CoWare
H/W-S/W cosim
Eagle-i  CoWare

H/W design
Fixed-point DSP H/W refinement
SystemC h/w synthesis
RTL signoff
Unicad
RTL-to-layout Tools

System integration
Aptix  Mentor Celaro

Central R&D
FlexWare Embedded System Tools

- System function
- System S/W Architecture
- Evaluation/Partitioning
  - FlexPerf: Performance eval.
- System H/W Architecture
  - FlexSim2: Arch. modelling

- S/W design
  - FlexCC Compiler
  - FlexGdb Debugger
  - FlexSim I/S simulation
  - FlexPerf S/W analysis

- Interface design
  - H/W-S/W cosim

- System integration

- H/W design
  - Unicad RTL-to-layout Tools
FlexWare1: Retargettable Embedded Software Development Tools

**Application-Specific Hardware**

- VHDL Functional description
- VHDL RTL description

**Processor Hardware**

- H/W-S/W co-simulation
- Object code
- Proc. model
- Compare
- VHDL RTL description of processor

**Embedded Software**

- C code
- FlexCC compiler
- FlexGdb debugger
- FlexSim model gen.

- FlexWare Processor Targetting Files

Central R&D
FlexWare1 Use for ST Processors

Audio Auto Periph. Industrial
- MMDSP+ (CC, Gdb)
- Sapphire AS-DSP (CC)
- Emerald AS-DSP (CC, Sim, Gdb)
- Orpheus DSP

Computer Communication Network
- Ivory AS-DSP (CC)
- Ruby2 AS-MCU (CC, Sim)
- D950 DSP (Sim)
- ST100 DSP/MCU (Sim, Perf)
- ST10 MCU (Sim proto)
- ARM7 RISC

Consumer
- ST20 MCU
- MMDSP (CC)
- D950 DSP (Sim)
- ST40 RISC
- Lx VLIW

Micro
- ST6 MCU (CC)
- ST7 MCU (Sim)
- Super7 MCU (Sim, Asm)
- ST9+ MCU (Gdb)

Central R&D
- MSQ AS-MCU (CC, Gdb, Sim)
- BSP AS-MCU (CC)
- VIP AS-VLIW (CC)

FlexWare tool used  Other tool
FlexWare2: Embedded Systems Development Tools

**Application-Specific Hardware**
- SystemC
- Functional description
- VHDL RTL description
- Cycle-based C model
- IDL2
  - FlexSim2 model gen.

**Processor Hardware**
- Object code
- Cycle-accurate Functional Proc. model
- VHDL RTL description of processor
- Compare
- FlexPerf performance analysis

**Embedded Software**
- C code
  - FlexCC2
  - FlexGdb debugger
  - FlexSim2
  - FlexSim model gen.
- FlexWare Processor Targetting Files
- Auto test
- Test vectors
- IDL2
  - IDL
- FlexSim2
  - Model gen.
- FlexCC
  - Compiler
  - Assembler

**H/W-S/W co-simulation**
- Cycle-accurate C model
FlexWare C Compilation: FlexCC

**Application-Specific Hardware**
- VHDL Functional description
- VHDL RTL description

**Processor Hardware**
- Object code
- Proc. model
- VHDL RTL description of processor
- Compare

**Embedded Software**
- C code
- FlexPerf performance analysis
- FlexCC compiler
- assembles
- FlexGdb debugger
- FlexSim model gen.

**HW-SW Methodologies**
Area Impact of Embedded S/W: ST set-top box audio example

MPEG2 audio

- SRAM 46.7%
- ROM 25.2%
- DSP core 21.0%
- Logic 7.1%

Karaoke

- SRAM 65.4%
- DSP core 17.1%
- Logic 17.6%

0.25 micron CMOS: 64% area is memory
0.18 micron CMOS: 70% area is memory (logic scales better)
**FlexCC: Retargettable CCompilation**

- Environment for the rapid development of ANSI C compilers
  - Covers large range of architectures
  - Quickly retargetable
  - Flexible enough to deal with particular features

- Main Usage: Application Specific Processors
  - DSP's: MMDSP, Ivory, Sapphire, Emerald
  - Micro-controllers: MSQ, HME, BSP, VIP
  - But also useable for standard processors

- Enabling technology for embedded processors
  - Key differentiator: development cost, time-to-market, dice area
  - Eliminates assembler programming
  - Strong influence on processor architects and designers
Use of FlexCC1 for MMDSP

- '95: MPEG2, Prologic
- '96: Dolby AC-3
- '97: MP3 satellite radio
- '98: Set-top box, DVD platform
- '99: MP3 walkman
- 2000: MP3 + GSM

FlexCC1 + debug info

FlexCC2
- High-level C code (e.g. Dolby, ETSI)
- Yields good results for control code
- Some inefficiencies for inner loops (DSP-oriented)
- Typical results on audio DSP:
  - Prologic: 20 MIPS
  - ETSI GSM EFR: 54 MIPS

- Profiling analysis of inner loops
- Inner loop recoding (s/w pipelining)
- Array to pointer conversion
- Compiler guidance pragmas
- Sample results (ANSI C):
  - Prologic: 8 MIPS 10 man-days
  - ETSI EFR: 24 MIPS 7 man-days
  - MLP (DVD) 60 MIPS 3 man-mth
Central R&D

FlexCC1  FlexCC2

Low-level C for DSP

C Control

FlexCC1

010101
110110
011100

Sapphire

MMDSP
MMDSP+

C DSP
C Control

FlexCC2

010101
110110

Emerald

MMDSP+

- Higher productivity
- Architecture independence
- Advanced Analysis
- Loop Optimizations
- Faster
- Smaller

STI
FlexCC2

- Main target: DSP's
  - Exploit parallelism
  - Architecture idiosyncracies

- Approach: ST added-value components on commercial ACE/CoSy compiler infrastructure

- Goals:
  - Maintain FlexCC1 code size
  - Improve productivity 2~4X
  - Best compilers for AS-DSP's
FlexCC2 Preliminary Results

Relative Performance 1.6X faster

![Bar chart showing relative performance of FlexCC1 and FlexCC2 for various benchmarks.]

- **FIR**: 3.0X speedup for FlexCC2
- **LatticeP**
- **LatticeZ**
- **Cascade1**
- **Cascade2**
- **FSM**

FlexCC2 is 1.6X faster than FlexCC1.
FlexCC2 Preliminary Results

Relative Code Size

Similar code size on average
Benchmarking FlexCC2 Beta-version

GSM EFR running on single-mac DSP (MMDSP+)
MIPS needed for real-time

- Optimizations ongoing:
  - Improved register allocation, global scheduling, array to ACU mapping
- Target for FlexCC2 = 24 MIPS
FlexWare Instruction-Set Simulation: FlexSim

Application-Specific Hardware
- VHDL Functional description
- VHDL RTL description

Processor Hardware
- H/W-S/W co-simulation
  - VHDL RTL description of processor
  - Proc. model
  - Object code
  - Compare

Embedded Software
- FlexCC compiler
- FlexGdb debugger
- FlexSim model gen.
- FlexPerf performance analysis
- C code
- FlexWare Processor Targetting Files

HW-SW Methodologies
Central R&D
FlexSim1: Instruction-set simulation

- Generation of high-performance C functional model of processor from abstract instruction-set specification
  - Bit and instruction-accurate (Version 1)
  - Short model development times: 2~8 man-weeks
  - Fast: >500 K instructions/sec
## FlexSim1 Targeting Experience

<table>
<thead>
<tr>
<th>Processor</th>
<th>Number of IDL lines</th>
<th>Effort (person-week)</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST6 8-bit MCU</td>
<td>900 lines</td>
<td>1.5 p-w</td>
<td>1.6 MIPs</td>
</tr>
<tr>
<td>ST7 8-bit MCU</td>
<td>1700 lines</td>
<td>2 p-w</td>
<td>675 kIPs</td>
</tr>
<tr>
<td>ST9+ 8/16-bit MCU</td>
<td>4000 lines</td>
<td>4 p-w</td>
<td>530 kIPs</td>
</tr>
<tr>
<td>Sapphire 16-bit AS-DSP</td>
<td>2200 lines</td>
<td>3 p-w</td>
<td>430 kIPs</td>
</tr>
<tr>
<td>D950 16-bit DSP</td>
<td>5500 lines</td>
<td>6~8 p-w</td>
<td>800 kIPs</td>
</tr>
<tr>
<td>ST10 16-bit MCU</td>
<td>5500 lines</td>
<td>6 p-w</td>
<td>570 kIPs</td>
</tr>
</tbody>
</table>

V.S. typical handwritten C++ model:
- 4~8 person-months effort
- ~10K instr./sec.

10X longer
50X slower
FlexSim1 V.S. FlexSim2

**FlexSim1**
- Functional model
  - 'Golden' reference
- Used for:
  - S/W tool validation
  - H/W design verification

**FlexSim2**
- Cycle-accurate model processor & peripherals
- Used for:
  - Architecture design & verification
  - H/W-S/W co-simulation
FlexWare Performance Analysis: FlexPerf

Application-Specific Hardware

- VHDL Functional description
- VHDL RTL description

Processor Hardware

- H/W-S/W co-simulation
- Proc. model
- VHDL RTL description of processor
- Compare
- FlexPerf performance analysis

Embedded Software

- C code
- FlexCC compiler
- assemblers
- FlexGdb debugger
- FlexSim model generator
- FlexWare Processor Targetting Files
FlexPerf Performance Analysis

Functionalities
- Application S/W analysis
- Processor use analysis
- Provides generic information collection and analysis services
- Flexible design
  ✓ Open API's to allow custom analysis module development
  ✓ Open API's to toolsets
  ✓ Processor retargettable

Helps designer achieve
- Reduced memory size & power consumption
- Higher real-time performance
- Reduced development time
FlexPerf Information Generation

- Application S/W code coverage
- Code size
- Results driven from C source
  - Code and variables
- Instruction-set analysis
  - Statistics on user-defined groups of instructions
- Resource analysis (mem, reg.)
  - By resource name
    - (incl. intervals for memory)
  - Resource access frequency
- Instruction count
FlexPerf: Instruction-Set Analysis

Percentage of executed instructions among group components

Effective number of executed instructions
Conclusion: FlexWare

- **FlexCC1**
  - Broad range of architectures
  - High-density control code

- **FlexCC2**
  - DSP-oriented
  - High-level C: DSP & control

- **FlexSim2**
  - High-speed architecture sim.

- **FlexGdb**
  - Gdb extension for DSP, MCU

- **FlexPerf**
  - Help explore new architectures
  - Guide processor usage
Outlook: Central R&D Roadmap for Platform Automation

Design Oriented Tools

- FlexSim2
  - micro-architecture modelling
  - Support for VLIW
- FlexCC2
  - High-performance S/W compiler
- FlexGdb
  - S/W debug (+ DSP features)
- FlexPerf
  - Performance Analysis
- Multi-processor platform automation
  - Support for multi-proc
  - Multi-processor platform verification
  - Multi-processor performance analysis
  - High-speed modelling
  - Support for real-time processing
  - Support for multimedia, caches
Outlook

- Acquisition of Nortel Semiconductor in May 2000
- New Architecture Platform Automation R&D Activity in Ottawa
- Focus on Telecom Applications

✓ For more info, contact pierre.paulin@st.com
Backup slides
**FlexGdb C source-level debugger**

- Link between C level description and compiled code

- Built on top of Gnu Gdb
  - Standard Gdb features
    - Standard RISC like architectures
    - Flat memory model

- FlexGdb Extensions for MCUs and DSPs
  - Complex memory models
  - Complex register structures and naming
  - DSP datatype support

- Integrated with FlexSim, FlexCC, FlexPerf
- Linked to DSP in-circuit emulator
- Beta link to Synopsys SystemStudio™ design tool
CoDesign Activity

- Focus on co-simulation
- Need of FlexSim2
  - Participation to FlexSim2 specification and development
- Develop additional modules and methodologies

System specification tools:
- COSSAP
- SPW
- MatLab

Architecture Simulation tools:
- CoWare N2C
- SystemC

FlexSim2

HW/SW co-simulation tools:
- Eaglei
- Seamless

Implementation:
- VHDL
- Verilog
- HW emulators
FlexSim ST100 model usage

**Specification:**
- 'Golden' functional reference
- Bones, CHESS model validation

**S/W, Tools, Applications:**
- S/W application development
- S/W tool validation
  - C compiler, debugger
  - Operating system

**Processor H/W design:**
- Test case generation
  - Linked w. Genesys
- Trace comparison
  - VHDL designers

**System integration (w. CoWare):**
- H/W-S/W co-simulation
- Co-emulation (Metasystems) ST100 + peripherals

**Customer executable specification**
### PIE: FlexSim Graphical Capture

**Selected Table:** SAPPH_instr

<table>
<thead>
<tr>
<th>Line Name</th>
<th>Encoding</th>
<th>Description</th>
<th>Assembler Syntax</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMAT1</td>
<td>0 MMU_13_bits MUL ALU</td>
<td>Multiply-Accumulate with Reg. Indirect +</td>
<td>&lt;M MUL &lt;MMU_13_bits&gt;;&lt;MMU_13_bits&gt;</td>
<td></td>
</tr>
<tr>
<td>FORMAT2</td>
<td>1 XMU_F2 YMU_F2 MUL ALU</td>
<td>Multiply-Accumulate with Reg. Indirect +</td>
<td>&lt;XMU_F2 &lt;YMU &lt;XMU_F2&gt; &lt;YMU_F2&gt;</td>
<td></td>
</tr>
<tr>
<td>FORMAT3</td>
<td>1 XMU_F3 YMU_F3 ALU_F3</td>
<td>Arithmetic/Logical/with Reg. Indirect +</td>
<td>&lt;XMU_F3 &lt;YMU &lt;XMU_F3&gt; &lt;YMU_F3&gt;</td>
<td></td>
</tr>
<tr>
<td>FORMAT4</td>
<td>immediate ALU_im Yin Dest</td>
<td>Arithmetic/Logical/with Immediate Data</td>
<td>&lt;ALU_imm &lt;Dest&gt;</td>
<td></td>
</tr>
<tr>
<td>FORMAT5</td>
<td>immediate MMU</td>
<td>Direct Register/Address Register</td>
<td>&lt;MMU&gt;</td>
<td></td>
</tr>
<tr>
<td>FORMAT6</td>
<td>IMM_CONTROL</td>
<td>Control / Address Register</td>
<td>&lt;IMM_CONTROL&gt;</td>
<td></td>
</tr>
</tbody>
</table>

Load: `/local/st6/WORK/in/Sapphire.pie`
FlexSim GUI Generation: Sapphire DSP