Configuring the Jazz DSP Core for Application Specific Requirements

Oz Levia
CTO

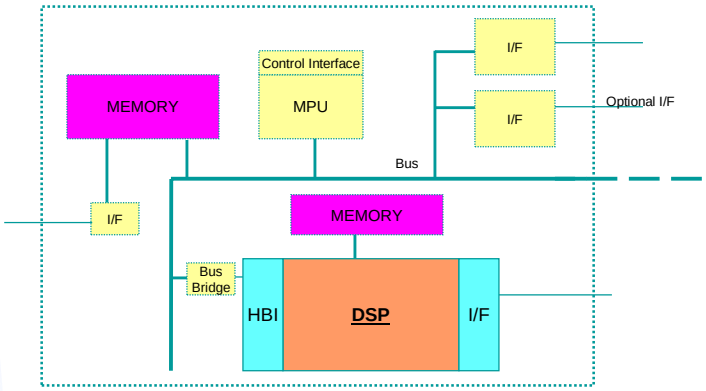
Multi processor SoC Summer School
France 2001

IC Platform for Data Oriented Applications

- Collection of Hardware and Software blocks and IC-IP
- MPU, DSP, Memory, I/O, I/F
- Embedded SW
  - Application 'payload'
  - Control, GUI, I/F
  - RTOS
- Key: Row Capabilities & Integration
  - Cost
  - Power
  - Function
  - Performance

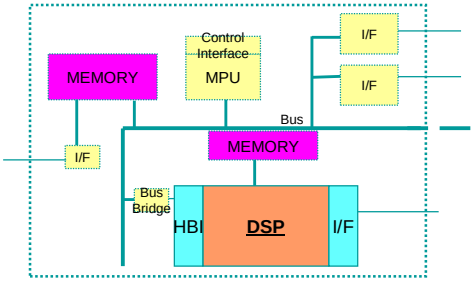
2

Platform



3

Platform: Requirements



- MPU: Fixed instruction set
  - Run protocols, user interface and existing code base
  - Low power
  - Master- Provide RTOS level control and interfaces
- DSP: Application Specific
  - Application performance
  - Power
  - Area and cost
- Memory: Bandwidth, power and area
- IO: Standard

4

Focus On The DSP

- In a platform, the DSP is responsible for data processing, the options are:
- Custom Logic
  - High performance but inflexible
- Standard programmable parts
  - Known entity
  - Compiler or Assembly code result in OK performance
  - But, one size does not fit all.
  - High cost.
- Embedded DSP
  - Similar to standard part,
  - "Hard" embedded DSP are Fab captive
- Multi Processor DSP
  - Little or no compiler support
- Other Solutions…

5

The Goal

Optimize a programmable DSP core(s) in a platform to match the requirements of an ***application***

Custom programmable DSP with Compiler and verification tools

Converge Quickly and predictably

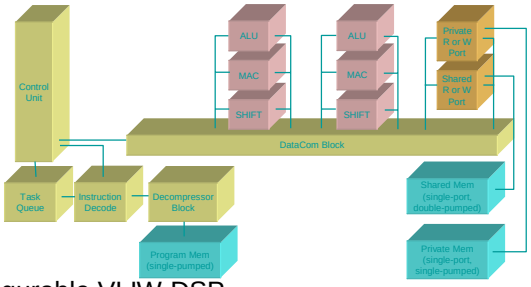
6

Requirements

- Low power DSP
  - Power consumption is an issue for embedded as well as standard parts
- High performance
  - High volume of data processing
- Programmable
  - Changing application needs
  - Time to market
  - Reuse of IC IP and Application code
- Flexible
  - Fit application size and bandwidth
- Easy to implement
  - Preserve time to market
- Development tools for application deployment
  - Development, verification, debug and optimization

7

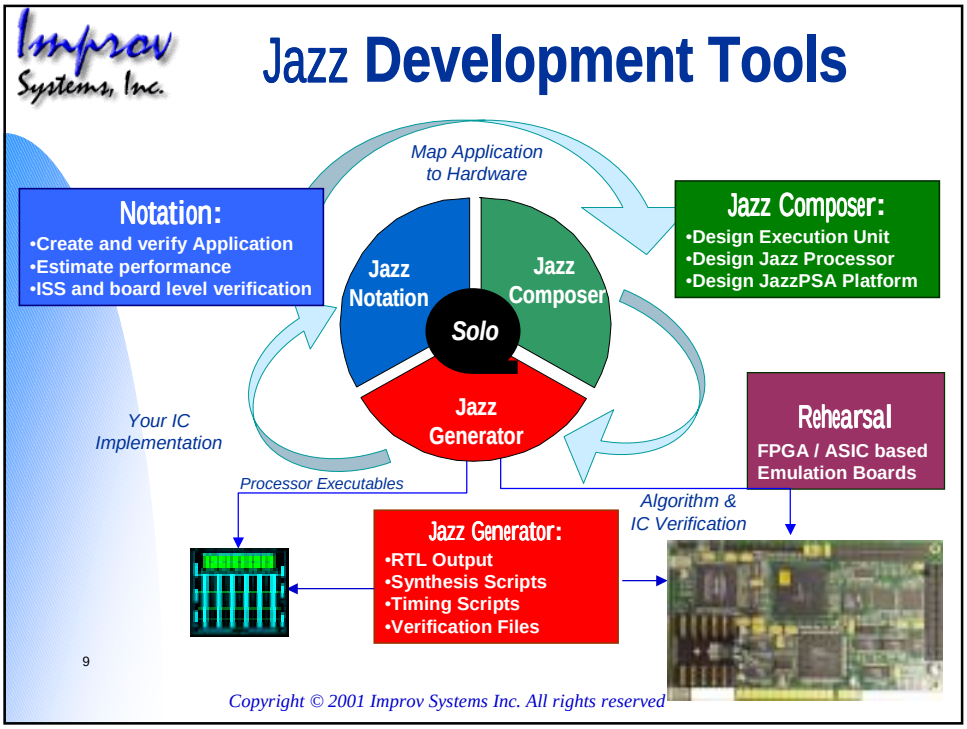
The Configurable Jazz DSP Processor



- Configurable VLIW DSP
  - Variable number of Micro Operations (MOP) per cycle
  - Variable memory bandwidth
  - Utilizes about 1 sq. mm of die in base configuration,
  - Standard configurations similar to standard DSPs
  - Utilizes less than 0.5 mW/MHz in 0.18 um @1.8v (TSMC)
  - High performance at low clock speed using VLIW approach
  - Requires no custom cells for physical design

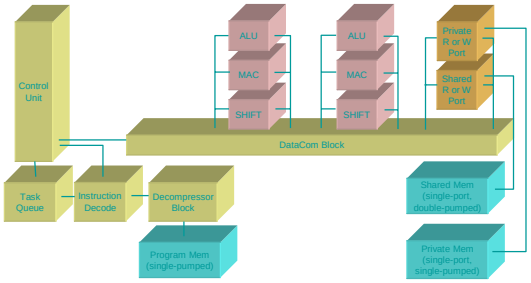
8

Jazz Development Tools



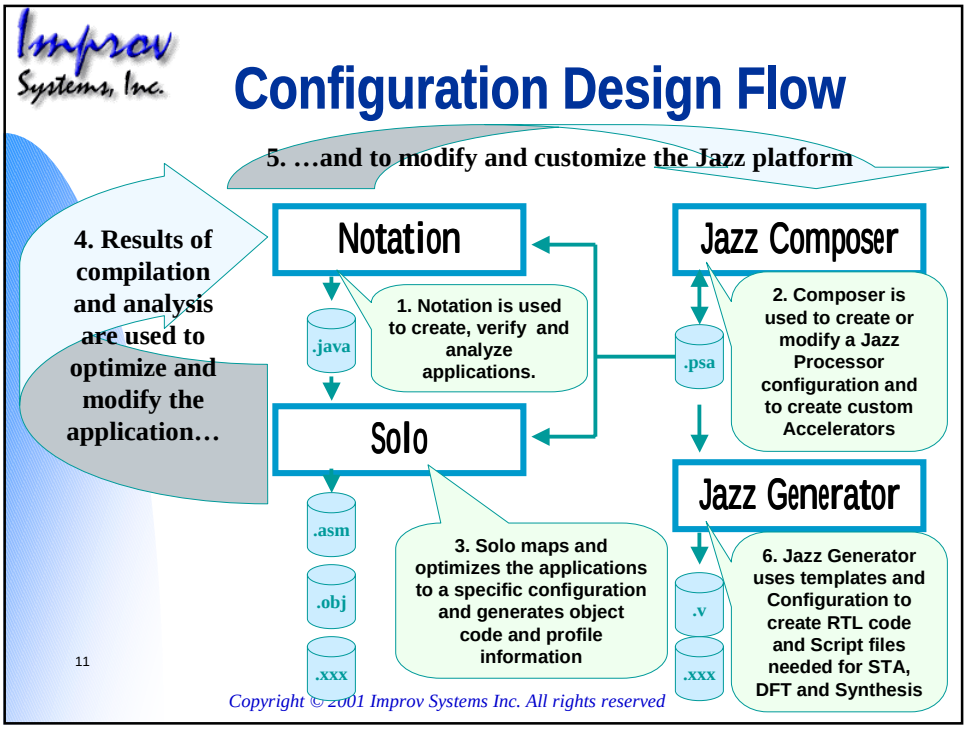
9

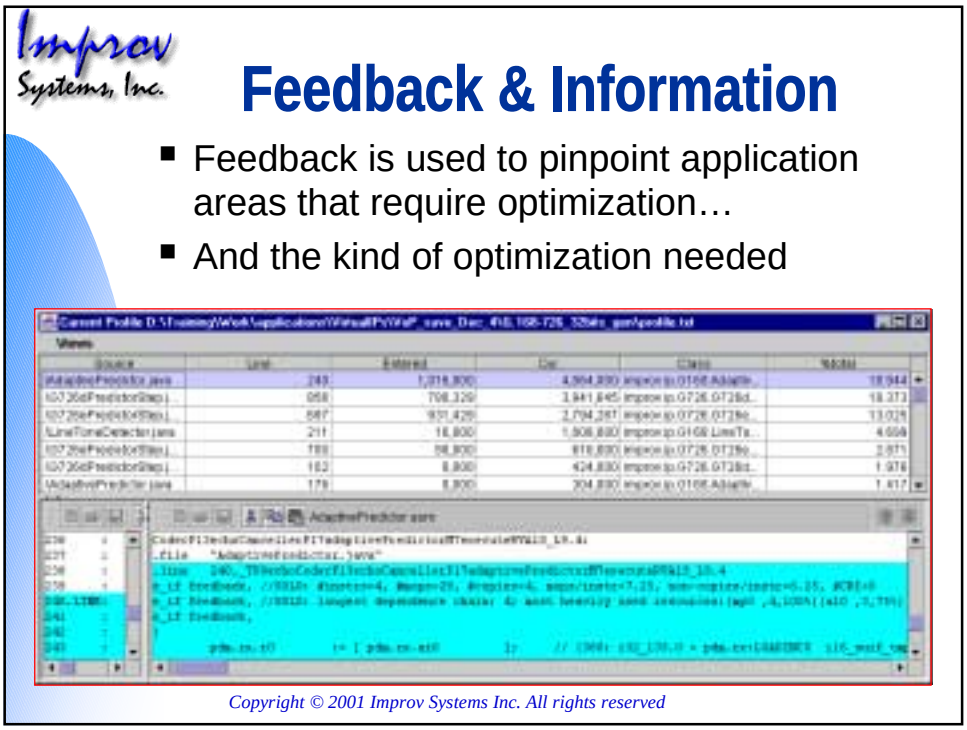
Configuring the Jazz Processor



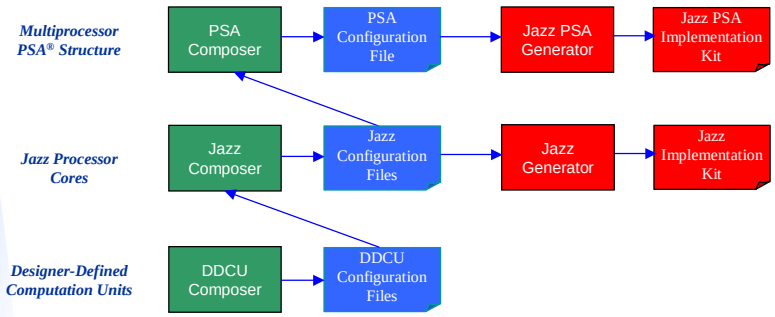
- Select the type and number of MOPs in a cycle
- Select the type and bandwidth of memory interface units (MIUs)
- 32-bit or 16-bit data path
- User defined routing of operands via a “DataComm” connectivity matrix
- Select a variable number of Registers
- Option to incorporate instruction compression
- Overlap Computational Units (CU’s) in any CU slot, and Memory Interface Units (MIU’s) in any MIU slot
- Insert custom- designer defined computational units (DDCU) for acceleration
- Multi-cycle and Pipeline Cus
- Configure interrupt capability

10


Improv Systems, Inc.
Configuration Design Flow
5. …and to modify and customize the Jazz platform
4. Results of compilation and analysis are used to optimize and modify the application…
Notation
Jazz Composer
.java
1. Notation is used to create, verify and analyze applications.
.psa
2. Composer is used to create or modify a Jazz Processor configuration and to create custom Accelerators
Solo
Jazz Generator
.asm
.obj
.xxx
3. Solo maps and optimizes the applications to a specific configuration and generates object code and profile information
.v
.xxx
6. Jazz Generator uses templates and Configuration to create RTL code and Script files needed for STA, DFT and Synthesis
11
Copyright © 2001 Improv Systems Inc. All rights reserved

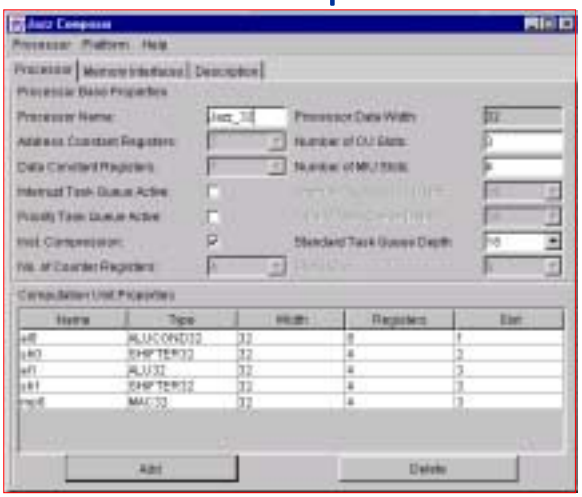

Improv Systems, Inc.
Feedback & Information
Feedback is used to pinpoint application areas that require optimization…
And the kind of optimization needed
Copyright © 2001 Improv Systems Inc. All rights reserved

Composer Hierarchical Configuration



13

The Composer Tool

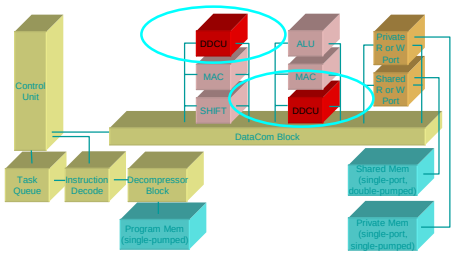


- GUI based tool for creating or modifying a DSP Jazz Processor

14

Typical Acceleration Blocks

- Designer Defined Computational Unit (DDCU)
  - Bit Stream oriented
  - Bit level operations
  - "Funny" Arithmetic
  - Multi-stage operations
    - IDCT
    - Filter
  - SIMD



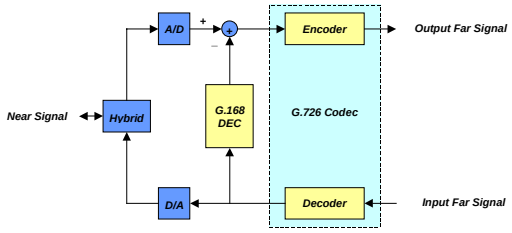
15

Example

- Starting point
  - Application: VoP- G.726 with G.168.
  - Target: Single Jazz DSP.
- Compilation & feedback
  - Baseline results
- Modify baseline *generic* resources
  - What to modify?
- Compilation & feedback
  - What was the effect of the change
- Design and add *custom* resources
  - Where to invest in optimization? Which optimization?
- Compilation & feedback
- …

16

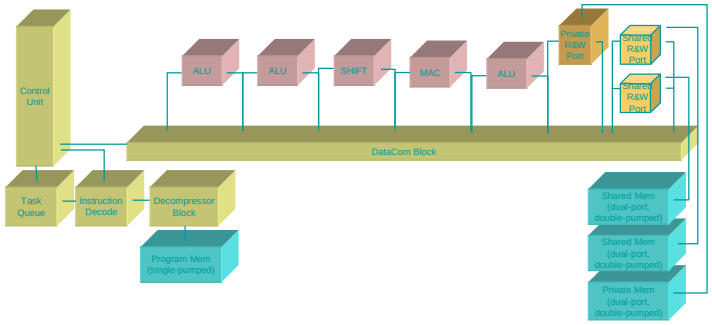
Voice over Packet Codec



- G.726 Vocoder and G.168 echo cancellation
- This is a portion of a larger application
  - Targeted for the Jazz DSP

17

Target Jazz DSP

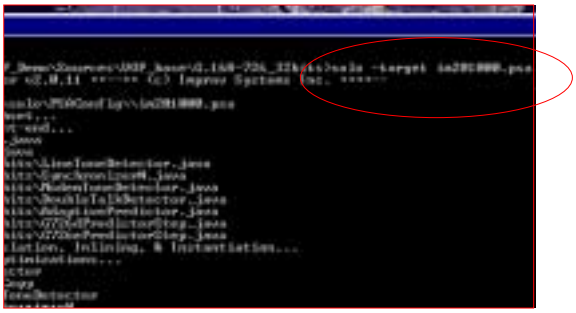


- VLIW Jazz DSP
  - Three ALU, one MAC, shift unit, four address units
  - All available at each cycle

18

First time out

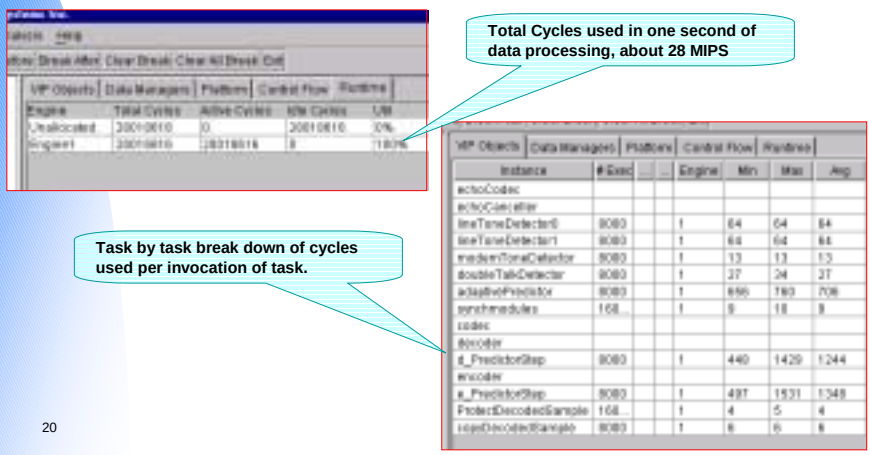
- Compile application to target:



19

Observe results

- Running 1 Second Of voice sample (far and near end)
- Use the information to get an idea of where to look for optimizations

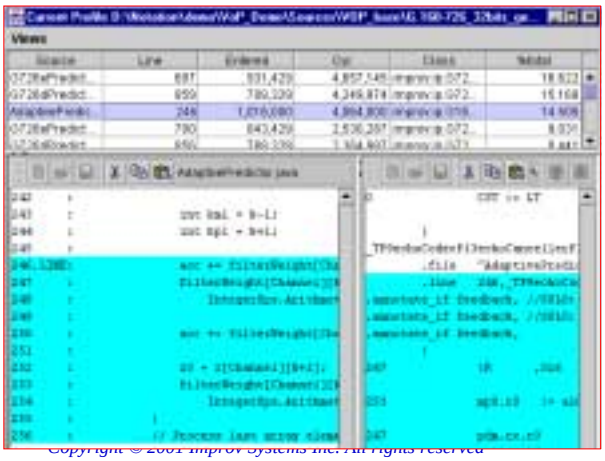


20

Improv Systems, Inc.

Analysis- Where?

- Need a clear picture of what can increase performance
- Look at tasks that are high percent users of cycles

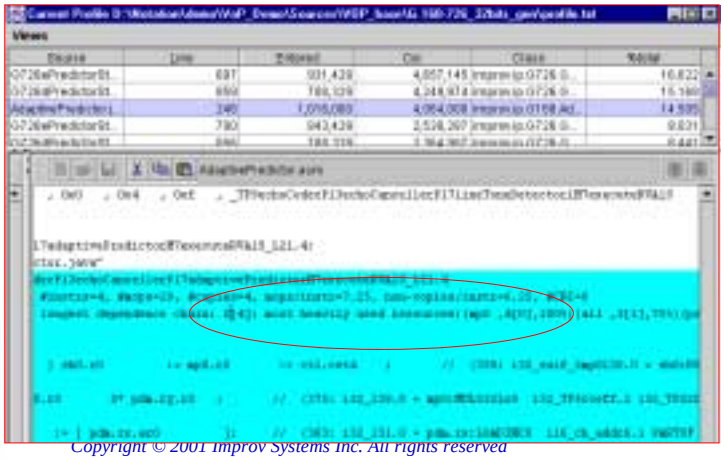


21

Improv Systems, Inc.

Analysis – What?

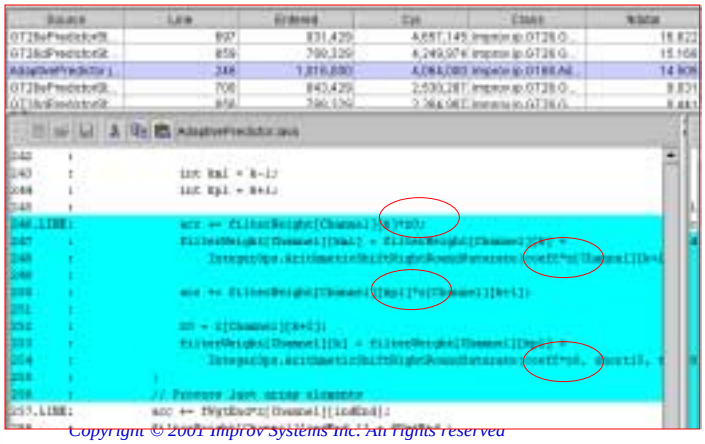
- Use information from the compiler: Which resources are constraining the application performance
- First look points to a MAC unit



22

Analysis – Confirm!

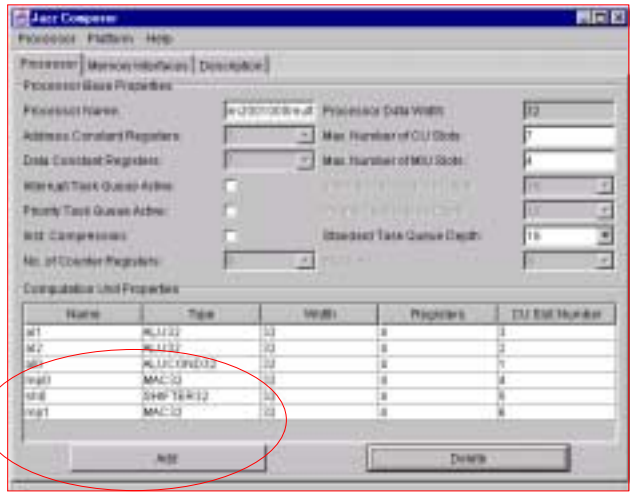
- Observe your application and confirm your analysis: Add a MAC Unit!



23

Modify the Jazz DSP

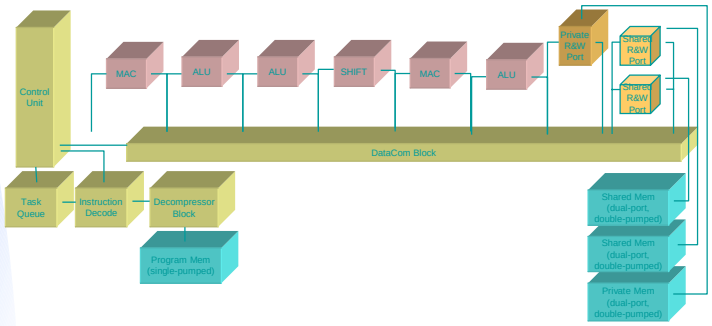
- Use Composer to add MAC



24

New Jazz DSP

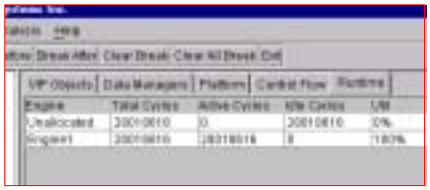
- One more MAC



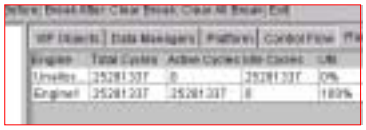
25

Compile and measure

- Compile application to new target
- Observe results- we gained 10%
- Before:



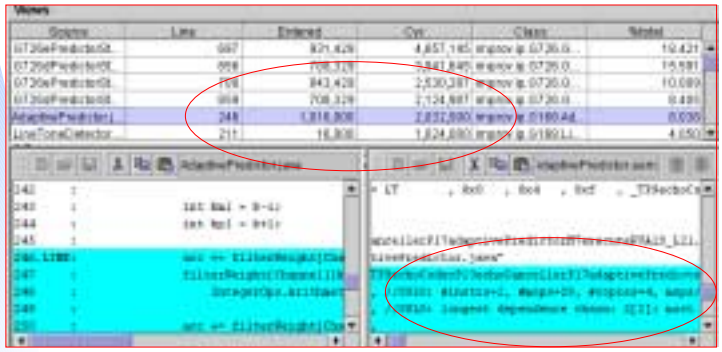
- After:



26

Did MAC unit really help

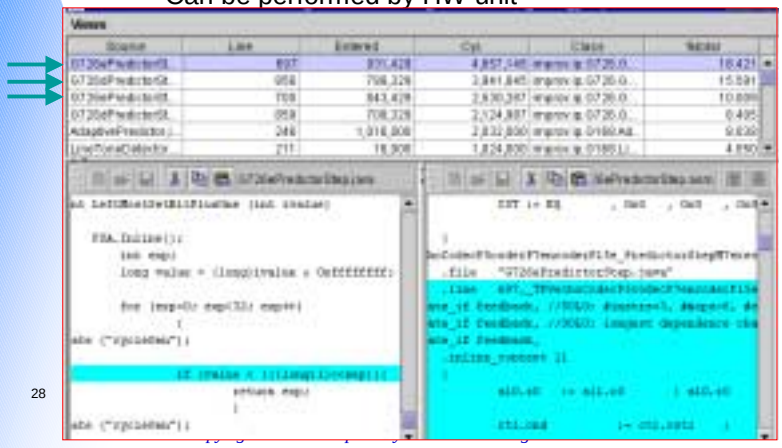
- Yes



27

Analysis- Where else can we improve?

- Function LMB+1 is called in many places
- Bit level operation is ‘emulated’ by ALU
- Can be performed by HW unit



28

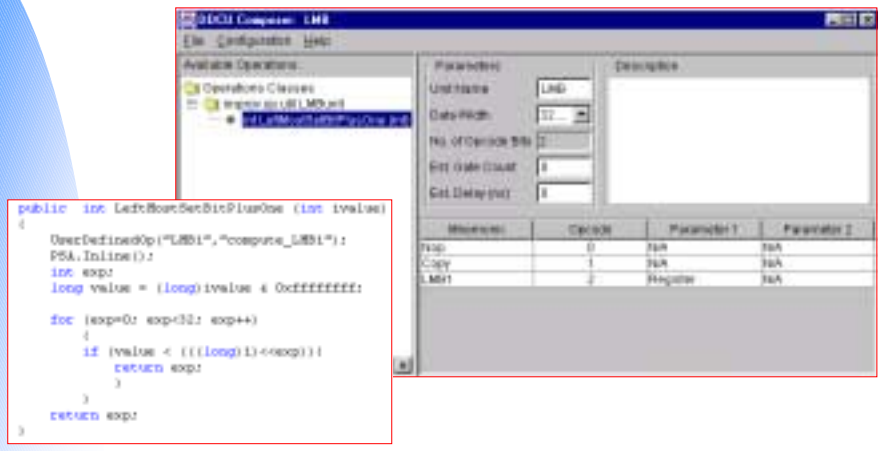
Create a DDCU

- DDCU: Designer Defined Computational Unit
- Function: ALU with special operations for LMB+1.
- Specifications
  - A class definition of the operation(s).
  - Each method defines an Op-code
  - DDCU can include multiple Op-codes
- Integration to application
  - Method call(s)
- Integration to Jazz
  - Configuration with composer
  - Add a CU

29

Creating a DDCU

- Use Composer to create DDCU



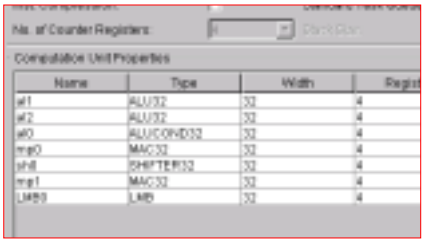
```
public int LeftMostSetBitPlusOne (int ivalue)
{
    UserDefinedOp("LMB1","compute_LMB1");
    PSA.Inline();
    int exp;
    long value = (long)ivalue & 0xffffffff;

    for (exp=0; exp<32; exp++)
        {
        if (value < (((long)1)<<exp))
            return exp;
            }
        }
    return exp;
}
```

30

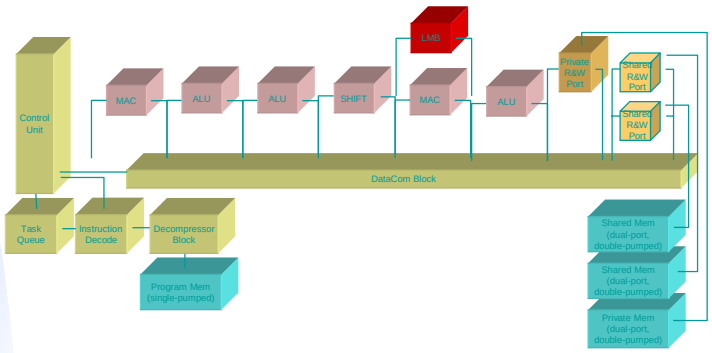
Add DDCU to Jazz DSP

- Add the new LMB unit into a Jazz DSP Processor
- New Jazz Processor now has Custom unit.



31

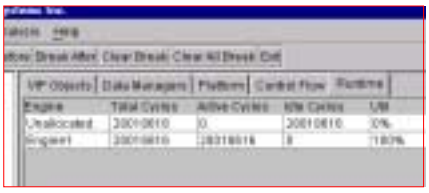
New Jazz DSP



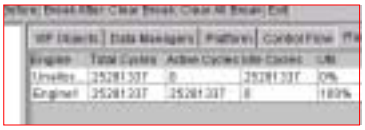
32

Compile and measure

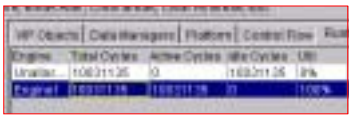
- Compile application to new target
- Observe results
- Before



- With MAC



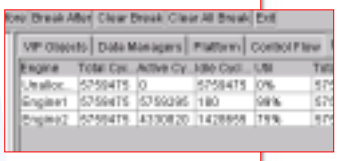
- With LMB & MAC

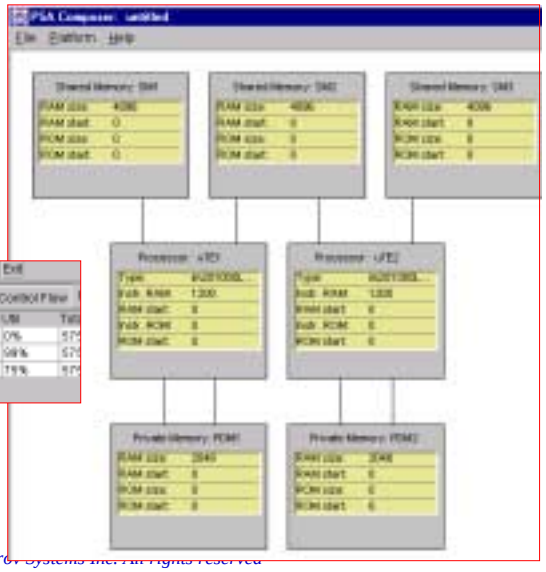


33

What next?

- One Processor is doing 'all it can'
- Lets try two processors
  - One processor with LMB
  - One processor without
- Results:

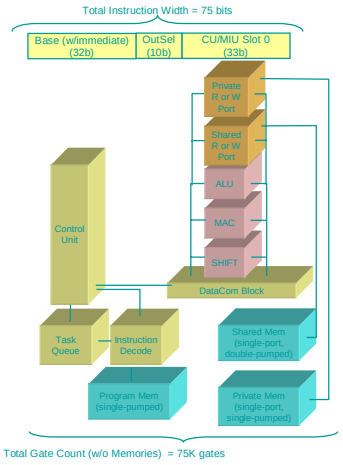




34

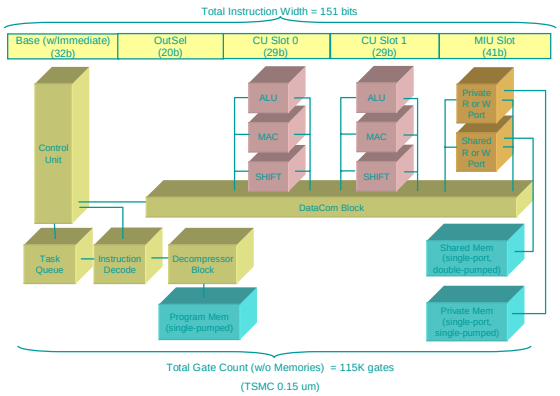
Jazz 2010 – Single MAC Processor

- Low End, Single instruction slot / multi-Computation Unit Configuration
- RISC-Like Processor with DSP Extensions
- Equivalent to DSPG Teak-lite, TI C54x, StarCore SC110, Intel/ADI XScale



35

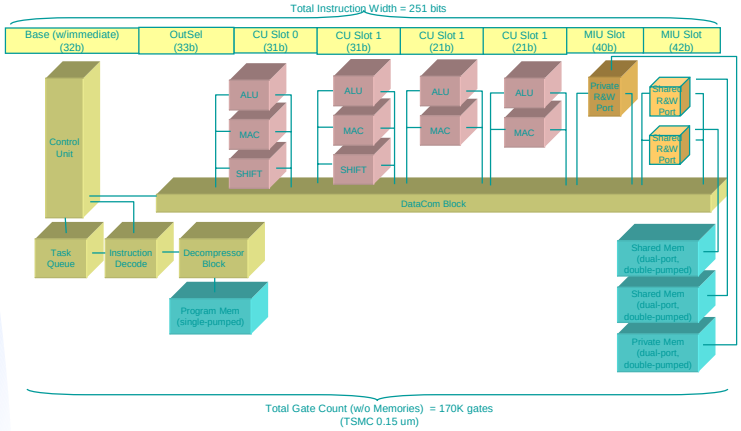
Jazz 2020 – DUAL MAC Processor



- Medium Range DSP for most new DSP applications in the market including VoP, Encryption, Edge Networking, 3G Wireless
- Comparable to TI C55x, StarCore SC120, 3DSP, DSPG Teak

36

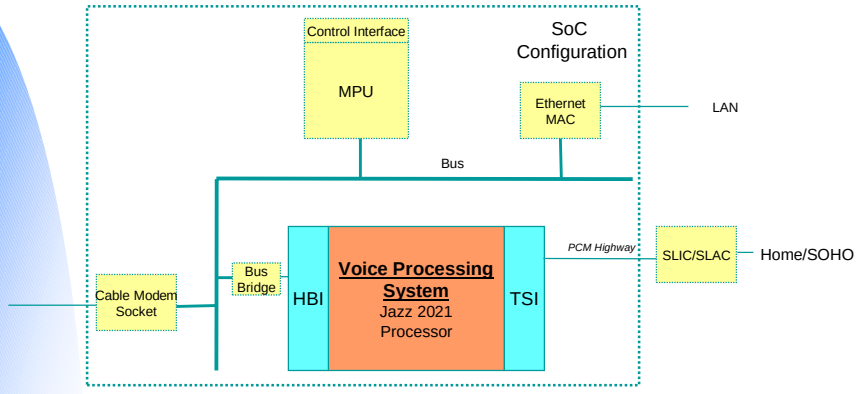
Jazz 2040 – Quad Mac Processor



- High end processor for performance intensive applications
- Competes with TI C62x, C64x, StarCore SC140, etc.

37

Acappella - PacketCable Reference Platform



- VPS could be a stand alone chip or used as a core integrated into an SoC

38

Improv Systems, Inc.

Conclusions

- VLIW DSP processor architecture with High performance
- Flexible architecture to meet the needs of applications
- Configuration file guide the tools chain in mapping to specific target
- Fast, tight loop: Compile, measure, change, Compile…
- Small changes in DSP can lead to great differences in performance
- It is critical to have he right feedback
  - Feedback is used to guide modifications
- The designer is the most important part of this loop

39