SONIC

A reconfigurable image processing architecture

A Joint project between
Sony Broadcast Europe &
Imperial College

Simon Haynes, Peter Cheung
Overview

- Introduction
- SONIC software architecture
- SONIC platform architecture
- SONIC Implementation - SONIC-1
- Current Status
- Summary
Introduction

Image processing has huge parallelism, and uses simple operations => Excellent candidate for hardware acceleration.

But...

Poor software models discourage usage of hardware.

Large data flow requirements.

SONIC Architecture - Simon Haynes
SONIC - Software Model I

SONIC uses software **Plug-Ins**

- Allows a wide variety of applications to use SONIC.
- Hardware & Software implementations.

*SONIC Architecture - Simon Haynes*
SONIC - Software Model II

SONIC Architecture - Simon Haynes
Beating the Bottleneck

Observation:
Image processing tasks often consists of several consecutive stages:

2-D FIR Filter constructed using 2 1-D FIR Filters

SONIC Architecture - Simon Haynes
Objectives

◆ Support the *Plug-In* software methodology.

◆ Allow simple implementation of ‘*Data Flow*’ operations

◆ Use PCI bus as efficiently as possible - Burst Mode transferal of Images.
SONIC Main Board Architecture

Video Bus(es)

Local Bus Controller (LBC)

Host Bus

Configuration Control, PIPE Select, and Interrupt Signals

PIPE Flow Start

PIPE Flow End

PIPE Flow Buses are shaded

SONIC Architecture - Simon Haynes
Plug In Processing Element

- PE performs processing
- PR Handles Image Transferal & Signal conditioning
PR Functionality

Generates data for the PE in the required form.

- Burst memory accesses
- PIPEFlow routing
- Data formatting (YCrCb etc.)
Using SONIC I
Using SONIC I
SONIC I - Implementation

Main Board Uses 2x Altera 10K50
PIPEs use Altera 10K20 (PR) and 10K70 (PE)

SONIC Architecture - Simon Haynes
Current Project Status

- Plug-Ins written for Adobe Premiere & In House Sony Software.
- 19 Tap 2-D Separable Filter Plug-In uses 1 PIPE
- Gives 4x Speed-up over software
- Working on further Plug-Ins
- Developing API
Summary

- Plug-In Software model gives good abstraction from hardware.
- PIPE architecture well suited to software plug-in design.
- PIPE PR gives hardware design flexibility.
- Easy future expansion - new PIPEs
What would we like to see in future Altera devices?

- Fast configuration - support for *true parallel* configuration.
- Partially reconfigurable devices (*macro blocks*?)
- Easy register read back capability, for improved debugging.
- PLLs / DLLs
- More gates!

*Sonic Architecture - Simon Haynes*