Designing with Software Defined Silicon
<table>
<thead>
<tr>
<th>Session</th>
<th>Duration</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>10 min</td>
<td>Welcome, agenda, XMOS overview</td>
</tr>
<tr>
<td>XS1-G technology</td>
<td>10 min</td>
<td>What is software defined silicon and how does it work?</td>
</tr>
<tr>
<td>Design tools</td>
<td>10 min</td>
<td>Introduction to software defined silicon design flow and the XMOS tools</td>
</tr>
<tr>
<td>First lab – “Hello port”</td>
<td>20 min</td>
<td>Design, build and debug a simple port example</td>
</tr>
<tr>
<td>Break</td>
<td>15 min</td>
<td>Talk 1:1 with XMOS staff</td>
</tr>
<tr>
<td>Using threads</td>
<td>10 min</td>
<td>What are threads and how do they work in a software defined system?</td>
</tr>
<tr>
<td>Second lab – “Multithreading”</td>
<td>35 min</td>
<td>Design, build and debug a multi-threaded example</td>
</tr>
<tr>
<td>Break</td>
<td>15 min</td>
<td>Talk 1:1 with XMOS staff</td>
</tr>
<tr>
<td>Applications for the XS1-G</td>
<td>10 min</td>
<td>Sample applications for XMOS chips</td>
</tr>
<tr>
<td>Application walk through</td>
<td>40 min</td>
<td>System specification and block diagram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Partitioning and mapping threads on the XS1-G XDK hardware demonstration</td>
</tr>
<tr>
<td>Summary</td>
<td>5 min</td>
<td>Conclusion and final messages</td>
</tr>
<tr>
<td>Total</td>
<td>3.0 hours</td>
<td></td>
</tr>
<tr>
<td>Your questions answered</td>
<td>1.0 hour</td>
<td>Complete further tutorials, find out more about XMOS technology or discuss your application with XMOS staff</td>
</tr>
</tbody>
</table>
XMOS Introduction
About XMOS

- Fabless Semiconductor Company
  - Founded in 2005 by experienced team
  - Based in Bristol, UK

- Next-generation programmable semiconductor technology: **Software Defined Silicon**
  - Groundbreaking 32-bit, multi-threaded, event-driven processor
  - Designs use a fast software design flow
  - 17 patents filed

*Custom hardware from a software design flow*
Electronics Design Challenges

- Electronic products rely on standards but...
  - They often change
  - Everyone uses them, so how do you differentiate?

- Software designers outnumber hardware engineers
  - High level language hardware flows have drawbacks
  - Processors and fast I/O haven’t mixed well

- Business demands flexible yet stable platforms
  - Maximise return on investment
  - Support fast turn-cycles and market changes

*This all just got easier...*
XS1-G Technology
The Ingredients of SDS

- Hardware multi-threaded processor
  - Efficient and simple MIPs sharing
  - Deterministic operation of each thread
- Event driven execution
  - Highly responsive
  - Thread execution (run/pause) driven by events
- Intelligent ports
  - Offloads each thread
  - Includes timing & external clock domain support
- Communications
  - Allows systems to be built from threads

*An real-time operating system kernel in hardware*
XCore Tile Overview

- Multi-threaded architecture
  - 8 threads per XCore
  - Event driven processing
  - Up to 400 MHz operation
- Mixed 16/32 bit ISA
  - High code density
- 64 I/O ports per core
  - Mix of 32, 16, 8, 4, 1 bit ports
  - External clock synchronisation
- Large integrated memory
  - 64K Bytes SRAM per tile
  - 8K Bytes OTP per tile

© 2008 XMOS Ltd.
Threads Explained

Up to 8 Threads

XCore Tile
- 8KB OTP
- 64KB SRAM
- JTAG TAP
- Input/Output Ports
- XLink Channel Ends

XCore Processor (8 threads)

100 Mb/s interfaces

State machines

Software tasks

Data processing

© 2008 XMOS Ltd.
Real-Time Performance

Interrupt servicing

- Save all registers
- Fetch ISR vector
- Execute Task B
- Clear request
- Restore registers

Event response

- Execute Task B
- Task B paused
- Event handled in the next thread cycle
- Respond to ‘n’ possible events using selection
- Use 100% of the resources with no performance loss

- All tasks in the XS1-G are high priority
- Event handled in the next thread cycle
- Respond to ‘n’ possible events using selection
- Use 100% of the resources with no performance loss
Event Sources and Thread State

- **Pins & Ports**: Compare with time or value, not =
- **Timers & Synchronisers**:
  - 10 ns Timer Compare with value =, >
  - Sync run / pause
- **XLinks**: Receive Or Transmit

Thread execution based events with next cycle response
I/O Port Features

- 64 I/O pins per XCore tile
  - Mix of 1, 4, 8, 16, 32 bit ports
  - Single cycle access
  - Up to 256 I/O pins in the XS1-G4

- Tightly coupled to core
  - I/O may generate events

- Abstracts hardware domain to the software domain
## I/O Advanced Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timed Output</strong></td>
<td>• Generate timed waveforms such as LCD, PWM, MII etc.</td>
</tr>
</tbody>
</table>
| **Time-stamped Input** | • Decode timed protocols  
• Reduces processor overhead |
| **Predicated Input** | • Thread waits for given pin state  
• Very efficient state machines |
| **Clocked port with data ready** | • Synchronise to external clocks  
• Support for throttled data |
| **Dedicated serialisation hardware** | • Serialise/deserialise data  
• Extends pin data rate beyond instruction rate |

© 2008 XMOS Ltd.
XLink Channel Interconnect

- Deterministic communications switch fabric
  - Connects threads, cores & chips
  - Thread communication abstracted to software
  - Automatically synchronises threads
  - Non blocking switch

Threads on same core
- Threads on different core
- Threads on different chip
## XS1-G Product Family

<table>
<thead>
<tr>
<th>Devices</th>
<th>XS1-G1</th>
<th>XS1-G2</th>
<th>XS1-G4</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCore Tiles</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Threads</td>
<td>8</td>
<td>16</td>
<td>32</td>
</tr>
<tr>
<td>Total MIPs</td>
<td>400</td>
<td>800</td>
<td>1600</td>
</tr>
<tr>
<td>Total SRAM</td>
<td>64 KBytes</td>
<td>128 KBytes</td>
<td>256 KBytes</td>
</tr>
<tr>
<td>Availability</td>
<td>Q1’09</td>
<td>1H’09</td>
<td>Now</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Device</th>
<th>TQFP64 0.8mm 12 x 12mm</th>
<th>PBGA144 0.8mm 11 x 11mm</th>
<th>PBGA336 0.8mm 17 x 17mm</th>
<th>PBGA512 0.8mm 20 x 20mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>XS1-G1</td>
<td>32*</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>XS1-G2</td>
<td>-</td>
<td>88</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>XS1-G4</td>
<td>-</td>
<td>88</td>
<td>176*</td>
<td>256</td>
</tr>
</tbody>
</table>

*Under Planning. Please contact XMOS for details.
XS1-G Design Tools
Design Flow

- The XMOS design and debug flow is similar to other embedded toolchains
  - Focus on C
- Adds language support to simplify tasks relating to concurrency and real-time control
  - XC
- Complete set of tools from design capture to advanced debugging
- Is accessible both online or on your desktop machine using downloadable tools
How does XC compare with C?

Support for:
- I/O with timing
- Communication
- Events
- Multiple threads and cores

XC looks and feels like C

Floating point
Pointer arithmetic
volatile

Use C, XC or a mixture of both

© 2008 XMOS Ltd.
Debugging and Profiling Tools

- **Debug**
  - XC, C and assembly code debugging for multithreaded/m multicore targets
  - Online using Java visual debugger & Instruction Set Simulator (ISS)
    - Connect to XS1-G chip coming soon
  - On your desktop using XMOS version of industry-standard GNU Debugger (XGDB)
    - Connect to cycle-based simulator or XS1-G chip via JTAG
    - GUI optional via Eclipse

- **Profiling**
  - High-level analysis, simplifies partitioning task
  - Online JAVA profiler using integrated ISS
    - Analyse core & thread activity, memory footprint, processor usage and ports
  - On desktop, analysis of waveforms using VCD output from cycle-based simulator
# XMOS Development Kits

<table>
<thead>
<tr>
<th>Item</th>
<th>XC-1 Development Kit</th>
<th>XDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMOS device</td>
<td>XS1-G4</td>
<td>XS1-G4</td>
</tr>
<tr>
<td>User I/O expansion</td>
<td>60 pins</td>
<td>128 pins + XLinks</td>
</tr>
<tr>
<td>Design tools</td>
<td>Included</td>
<td>Included</td>
</tr>
<tr>
<td>LEDs and switches</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Audio</td>
<td>On board speaker</td>
<td>96KHz Stereo codec w/3.5mm jacks</td>
</tr>
<tr>
<td>Ethernet</td>
<td>-</td>
<td>10/100Mbit</td>
</tr>
<tr>
<td>USB</td>
<td>-</td>
<td>USB 2.0 PHY</td>
</tr>
<tr>
<td>Video</td>
<td>-</td>
<td>¼ VGA Colour touch screen TFT</td>
</tr>
<tr>
<td>PSU</td>
<td>Via USB</td>
<td>12V SMPS included</td>
</tr>
<tr>
<td>Price</td>
<td>$99</td>
<td>$999</td>
</tr>
</tbody>
</table>

*Development kits for software and hardware designers*
Designing with SDS - lab 1
Hello Port!

- Toggle an LED and write a UART transmit function in XC:

- Focus on:
  - ports
  - timers
Using threads as building blocks
Threads Explained

Software tasks

```
#include "binarySearch.h"

// From example in code
int search = 1;
int test = table[0] = 1;
int result = 0;
int value = 0;

while (++value < (found ? found : found >> 1)) {
    if (search == value / 2)
        return value;
    if (search < value)
        return search;
    if (search > value)
        return value;
}
```

Data processing

XCore Tile
- 8KB OTP
- 64KB SRAM
- JTAG TAP
- Input/Output Ports
- XLink Channel Ends

XCore Processor
(8 threads)

Thread

100 Mb/s interfaces

State machines
Example: Partitioning a UART

![Diagram showing UART components and ports with connections]

- **Port 1**: DataBusIF, Data Bus Buffer, Select and Control Logic
- **Port 2**: I/O R, I/O N, CS, Reset, A0, A1, A2, INTR
- **Port 7**: D0:D7
- **Port 3**: Sout
- **Port 4**: nRTS, nDTR
- **Port 5**: Sin, nDCD, nCTS, RI
- **Port 6**: Port 3, Port 4, Port 5, Port 6

*XLink Channels*
Mandelbrot Demo Thread Diagram

Touch screen X-Y data ports
3 x 1-bit IN
3 x 1-bit OUT

Mandelbrot calculation

T_0

Mandelbrot calculation

T_1
T_2

T_3 T_4

Buffer Control

T_5

LCD Physical

T_6

RGB data + LCD timing
16-bit
2 x 1-bit OUT

LCD Ports

Touch screen & Manager

T_7

SRAM Controller

SRAM Ports

Address, Data & Control
19-bit OUT
8-bit BIDIR
3 x 1-bit OUT

https://designing.xmos.com/online-engineering/project/instantiate/9369
Designing with SDS - lab 2
Hello Thread!

- Focus on:
  - threads
  - channels
  - select
Hello Thread!

- Focus on:
  - threads
  - channels
  - select
Debug and Analysis

Using GDB for multicore
  • Breakpoints
  • Single stepping

Optional analysis using online tools:
  • Profiling
  • Communication
  • VCD
Applications for the XS1-G
XMOS Usage Scenarios

Usage Case 1
• Intelligent Bridge

Usage Case 2
• I/O Expansion or companion chip

Usage Case 3
• XMOS based ASSP
Networked Audio Breakout Box

- Highly customisable Audio Box
  - Add any blend of interfaces
  - Easy changes during lifecycle
- Software-based design
  - Software-based stacks for changing standards (802.1as)
  - Audio correction DSP from C
  - Proprietary algorithms secured in XMOS OTP ROM
- Ideal for production
  - XS1-G4 for 7.1 in/out, S/PDIF in/out, equalization, control
  - XS1-G1 for network speakers
  - Platform design with single code base across entire range

<table>
<thead>
<tr>
<th>Task</th>
<th>Threads</th>
<th>RAM (Bytes)</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
<td>6</td>
<td>64K</td>
<td>14</td>
</tr>
<tr>
<td>I8S</td>
<td>7</td>
<td>4K</td>
<td>12</td>
</tr>
<tr>
<td>S/PDIF</td>
<td>5</td>
<td>4K</td>
<td>2</td>
</tr>
<tr>
<td>Control + Debug</td>
<td>4</td>
<td>16K</td>
<td>8</td>
</tr>
<tr>
<td>Totals</td>
<td>22</td>
<td>88K</td>
<td>36</td>
</tr>
</tbody>
</table>
LED Tile Display Controller

- High fidelity display controller
  - Performance to match displays
  - Scales to any size LED tile, and any arrangement of tiles

- Software-based design
  - Colour and luminance correction in software
  - Local ambient corrections (e.g. shadow management)

- Ideal for production
  - XS1-G4 for large tiles, XS1-G2/G1 for smaller/consumer
  - Safe & simple field upgrades
  - XLink interconnect simplifies networking of tiles

<table>
<thead>
<tr>
<th>Task</th>
<th>Threads</th>
<th>RAM (Bytes)</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWM generation</td>
<td>2</td>
<td>4K</td>
<td>30</td>
</tr>
<tr>
<td>Image Processing</td>
<td>8</td>
<td>16K</td>
<td>0</td>
</tr>
<tr>
<td>Control &amp; tile communications</td>
<td>3</td>
<td>8K</td>
<td>6</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td>13</td>
<td>28K</td>
<td>36</td>
</tr>
</tbody>
</table>
Intelligent I/O Aggregation

- XLink-based interconnector
  - Reduces I/O physical count
  - Increases mechanical reliability
  - Ideal for hinged devices

- Software-based design
  - n-channels in 1 XLink channel
  - Easy channel based coding

- Ideal for production
  - Reduces hardware costs
  - Requires no transceivers
  - Uses two XS1-G1 devices
  - Reduces field failure costs

<table>
<thead>
<tr>
<th>Task (1 end)</th>
<th>Threads</th>
<th>RAM (Bytes)</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td>5</td>
<td>6K</td>
<td>19</td>
</tr>
<tr>
<td>Control</td>
<td>1</td>
<td>4K</td>
<td>0</td>
</tr>
<tr>
<td>XLink Port</td>
<td>0</td>
<td>0K</td>
<td>4</td>
</tr>
<tr>
<td>Totals</td>
<td>6</td>
<td>12K</td>
<td>23</td>
</tr>
</tbody>
</table>

© 2008 XMOS Ltd.
## Reference Software - Interfaces

<table>
<thead>
<tr>
<th>Hardware Task</th>
<th>Threads</th>
<th>Memory, K</th>
<th>I/O</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>UART Full Duplex</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>now</td>
</tr>
<tr>
<td>4ch UART Full Duplex</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>now</td>
</tr>
<tr>
<td>I2C Master or Slave</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>now</td>
</tr>
<tr>
<td>10/100 Ethernet MAC &amp; MII</td>
<td>4</td>
<td>4</td>
<td>14</td>
<td>now</td>
</tr>
<tr>
<td>10/100 Ethernet AV MAC &amp; MII</td>
<td>7</td>
<td>16</td>
<td>14</td>
<td>now</td>
</tr>
<tr>
<td>2x Transport Stream Serial / Parallel</td>
<td>3</td>
<td>4</td>
<td>5 / 12</td>
<td>now</td>
</tr>
<tr>
<td>USB 2.0 Device ULPI 480Mbps</td>
<td>4 of 4</td>
<td>8</td>
<td>12</td>
<td>Q1 2009</td>
</tr>
<tr>
<td>USB 2.0 Device ULPI 12Mbps</td>
<td>4 of 8</td>
<td>8</td>
<td>12</td>
<td>Q1 2009</td>
</tr>
<tr>
<td>SRAM Interface 20MHz 16bit data</td>
<td>1</td>
<td>1</td>
<td>40</td>
<td>now</td>
</tr>
<tr>
<td>I2S Master or Slave</td>
<td>1</td>
<td>0.5</td>
<td>3</td>
<td>now</td>
</tr>
<tr>
<td>SPI Master or Slave</td>
<td>1</td>
<td>0.5</td>
<td>4</td>
<td>now</td>
</tr>
<tr>
<td>LCDC - 1/4 VGA resolution</td>
<td>1</td>
<td>2 + framebuffer</td>
<td>19</td>
<td>now</td>
</tr>
<tr>
<td>MMC/SD interface</td>
<td>2</td>
<td>8</td>
<td>4</td>
<td>now</td>
</tr>
</tbody>
</table>
# Reference Software - Stacks

<table>
<thead>
<tr>
<th>Software component</th>
<th>Thread</th>
<th>Memory, KB</th>
<th>Source</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>UDP Stack</td>
<td>1</td>
<td>16</td>
<td>XMOS</td>
<td>Now</td>
</tr>
<tr>
<td>TCP/IP Stack</td>
<td>1</td>
<td>30</td>
<td>3rd Party</td>
<td>Now</td>
</tr>
<tr>
<td>AES (128 bit, 10 Mbit/s)</td>
<td>1</td>
<td>8</td>
<td>XMOS</td>
<td>Now</td>
</tr>
<tr>
<td>AES (128 bit, 20 Mbit/s)</td>
<td>2</td>
<td>10</td>
<td>XMOS</td>
<td>Now</td>
</tr>
<tr>
<td>USB 2.0 Class Driver</td>
<td>1</td>
<td>40</td>
<td>TBD</td>
<td>Q1 2009</td>
</tr>
<tr>
<td>FAT16 File System</td>
<td>1&lt;</td>
<td>6</td>
<td>XMOS</td>
<td>Now</td>
</tr>
<tr>
<td>FIR Filter</td>
<td>1 - 8</td>
<td>0.5 - 4</td>
<td>XMOS</td>
<td>Now</td>
</tr>
<tr>
<td>MP3 Decode (44KHz stereo)</td>
<td>1</td>
<td>60</td>
<td>XMOS</td>
<td>Q4 2008</td>
</tr>
<tr>
<td>Motion JPEG</td>
<td>TBD</td>
<td>TBD</td>
<td>XMOS</td>
<td>Q4 2008</td>
</tr>
</tbody>
</table>
# Design Goals

**• Real-time Audio Filter with Ethernet AV input**

<table>
<thead>
<tr>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>• &gt; 50µs digital delay using 32-bank filter</td>
</tr>
<tr>
<td>• Touch screen controlled filter</td>
</tr>
<tr>
<td>• Real-time frequency display in &amp; out</td>
</tr>
<tr>
<td>• Ethernet AV input</td>
</tr>
<tr>
<td>• Screen buffer-less display driver</td>
</tr>
<tr>
<td>• 512-point FFT completed in 3ms for real time display</td>
</tr>
<tr>
<td>• HiFi yet flexible Implementation</td>
</tr>
<tr>
<td>• Entire system to fit in a XS1-G4</td>
</tr>
</tbody>
</table>
System Partitioning

Key
- Thread
- Channel
- Port

Ethernet PHY
- RJ45
- MII Rx
- MII Tx
- MAC
- 1588
- TCP/IP
- Buffer
- AVB
- Buffer
- L filter
- R filter
- Touch IC interface
- Touch screen IC
- LCD display
- Audio DAC
- Buffer
- FFT
- LCD render
- LCD control
- Filtered
- Unfiltered
- Keyboard scan and control

© 2008 XMOS Ltd.
Summary
## XMOS Roadmap

<table>
<thead>
<tr>
<th>Tools</th>
<th>Chips</th>
<th>Boards &amp; Kits</th>
<th>Q3 2008</th>
<th>Q4 2008</th>
<th>Q1 2009</th>
<th>Q2 2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools first release:</td>
<td>XS1-G4 Samples (ES)</td>
<td>XS1-G Devkit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● XC compiler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● C compiler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Simulator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Debugger</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Visual profiler</td>
<td>XS1-G4 Production</td>
<td>XC1-Card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Web tools</td>
<td></td>
<td>XS1-LED</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● C++ compiler</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Testbench capability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Power analysis</td>
<td>XS1-G1, G2 Samples (ES)</td>
<td>XS1 Robotics kit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Timing analyser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● System builder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Function wizard</td>
<td>XS1-G1, G2 Production</td>
<td>XS1 Industrial kit</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Virtual thread flow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>● Collaborative design</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

© 2008 XMOS Ltd.
Xlinkers

- Mission
  - Collect a large number of open source designs for XMOS Devices
  - Allow discussion and sharing of code and projects
  - To create a marketing free environment
  - Future: Conferences

- Why use Xlinkers
  - Support from a community of users of XMOS technology
  - To “Link” up with like minded individuals

http://www.xlinkers.org
Xlinkers Future

- Intentions
  1) Seed the community
  2) Find some community members to join the committee
  3) Grow to a sustainable size
  4) Give over control of the community
- The XMOS Advent Calendar
  - 24 code examples for the 24 days of advent
  - Interesting or entertaining uses of XC-1s, XDKs or XC
  - Published on the front of the xmos.com website
  - Submitted through the Xlinkers code upload

Hope to see you there!
Summary

- Differentiate your product using re-configurable silicon
  - Ultimate flexibility with a fully programmable device
  - Support multiple variants from a single design, or retrofit
  - No need to compromise on ‘best-fit’ standard product

- Develop it easily
  - 100% Software implementation
  - Fast development and revisions
  - Library of example functions

- Do it at a low cost
  - Lower cost than any other programmable technology
  - Development effort and time reduced with HLL implementation
  - Integrate multiple functions into the SDS device

Software Defined Silicon
Thankyou!

Your Questions?