

Technical overview of Milkymist SoC

Revision E (Mar. 2011)

Sébastien Bourdeauducq

1 Introduction

The MilkymistTM project [7] develops a stand-alone device in a small form factor that is capable of rendering MilkDrop-esque visual effects [13] in real time, with a high level of interaction with many sensors and using live audio and video streams as a base.

The flexibility of the FPGA used as a central component enables advanced users to modify the design, and also permits compact integration of many interfaces (Ethernet, USB, MIDI, DMX512, IR remote, video input), making MilkymistTM a platform of choice for the mobile VJ.

But MilkymistTM is more than a visual synthesizer - it is also developing and maintaining one of the leading open source system-on-chip designs. It is today the fastest open source system-on-chip capable of running uClinux, and it comes with an extensive set of features and graphics accelerators.

The IP cores that make up the system-on-chip are entirely written in open source synthesizable Verilog HDL with extensive use of logic inference to ease porting to various FPGA and ASIC technologies. They come with test benches and documentation. This makes MilkymistTM a great library of re-usable logic cores to serve as a base for other open source hardware.

This paper gives a technical overview of the system-on-chip and its environment.

2 System architecture

The block diagram of the complete system-on-chip is given in Figure 2. The complete system is written in synthesizable Verilog HDL.

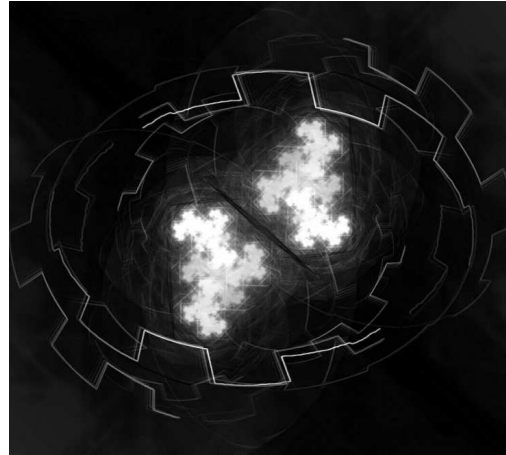


Fig. 1: Sample video output from MilkDrop

Its components are detailed below.

2.1 SoC interconnect

The Milkymist system-on-chip uses three different kind of buses :

- WISHBONE [9] as a general purpose bus around the CPU core.
- a custom “CSR” bus [1] used to access configuration and status registers of peripherals. It is simpler than WISHBONE; it does not support variable latency and the address decoding is simplified.
- a custom “FastMemoryLink” (FML) bus [2] which is pipelined and burst-oriented for efficient DRAM access.

By removing the need for logic that is only required to comply with a too general bus specification, the use of these specific buses reduce the hardware design effort and improve resource efficiency.

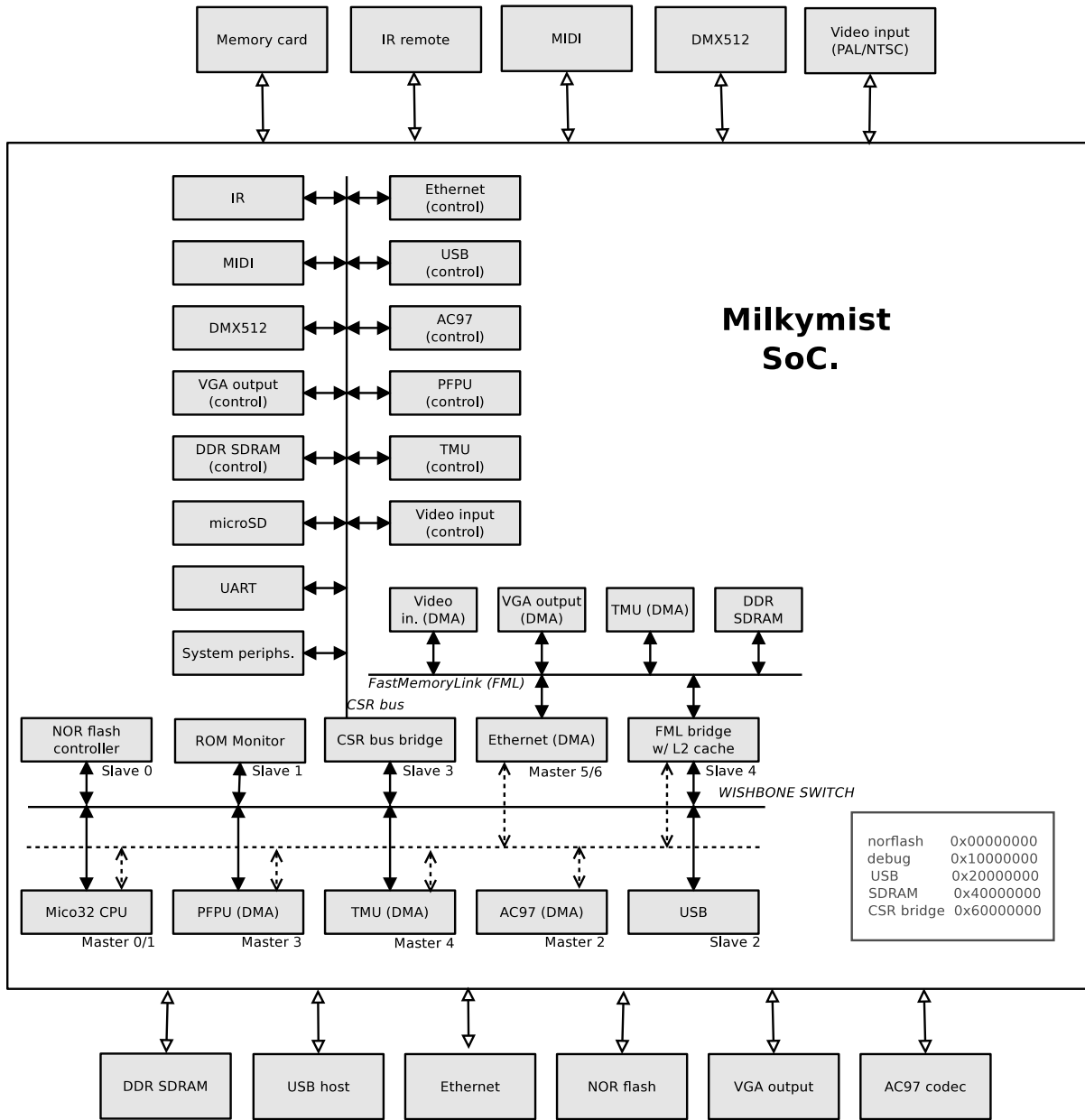


Fig. 2: SoC block diagram

2.2 Building blocks

2.2.1 Base system

The base system is made up of a LatticeMico32 CPU core [8], on-chip SRAM, off-chip Flash, an UART for printing debug messages, general-purpose I/O ports, timers, and interrupt controller.

The LatticeMico32 core can execute uClinux [12], or be programmed like a microcontroller, without operating system. Early versions of the system-on-chip used AEMB [16] instead of LatticeMico32. It has been replaced because of performance and software support complexity reasons.

These make up a basic system that is capable of executing software and communicate with the outside world. On top of this system, special peripherals and accelerators are added.

2.2.2 Memory interface

The system-on-chip is equipped with a custom DDR SDRAM controller [3], supporting a Fast-MemoryLink interface.

The DDR SDRAM data bus width is 32 bits and is running at up to 100MHz, delivering a peak (ideal) memory bandwidth of 6.4Gbps.

The memory controller is fully synchronous (the SDRAM clock is the system clock) to avoid clock domain crossing delays and reduce the overall memory latency.

It is a “page mode” controller, which leaves DRAM pages open after an access on the chance that the next access will be on the same page. This has been shown to be fruitful in most cases [14].

Memory latencies are further reduced by the use of pipelined transfers on the FML bus.

2.2.3 VGA output

The system-on-chip directly drives the H/V synchronization pins of the VGA interface and a video DAC that generates the red, green and blue analog signals.

The framebuffer is read from DRAM using the FML interface directly.

To cope with the hard realtime constraint of the video signal generation, the VGA controller

contains a FIFO which hides the memory latencies.

The framebuffer uses a simple progressive scan 16bpp RGB565 schema. The controller supports multiple buffering and synchronizes the switching between the framebuffers with the vertical blanking intervals in order to prevent drawing artifacts.

2.2.4 Texture mapping unit (TMU)

The unit maps a texture on a rectangular surface with texture coordinates defined on a grid of control points. It supports bilinear filtering, texture wrapping, alpha blending, additive drawing, and chroma keying.

To implement MilkDrop at a good frame rate, this becomes a very computation and memory intensive process. The implementation is heavily parallel, and is directly connected to the FML bus to achieve memory bandwidth constraints [4] [6].

2.2.5 Programmable floating point unit (PFPU)

The PFPU [5] is a floating point coprocessor, whose primary purpose is generating vertex data when implementing MilkDrop. It is similar to the vertex shader of traditional graphics processing units.

It is a pipelined VLIW processor with all the scheduling done by the compiler. This radical approach enables a very efficient use of the FPGA resources. Loop structures are not programmable, which limits the use case to evaluating mathematical expressions.

2.2.6 Audio I/O

The audio controller interfaces the system-on-chip to industry-standard and cheap AC97 codecs.

It supports full-duplex audio operation at a 16-bit 48kHz sample rate and access to the AC97 codec registers.

2.2.7 Memory card

The system is equipped with a memory card controller compatible with the popular memory

cards, which are used to store firmware, user media and data.

2.2.8 Ethernet

The Milkymist SoC can connect to industry-standard Ethernet PHYs to enable TCP/IP network connectivity and specifically the OpenSoundControl protocol which supersedes MIDI for the connection of electronic instruments.

2.2.9 USB

The system-on-chip integrates a protocol stack compatible with full-speed host USB. The SoC connects directly to USB transceivers to interface the ports.

The USB-compatible ports allow connection of keyboards, mice, USB sticks, wireless network cards, etc.

2.2.10 DMX512 and MIDI

These two interfaces are common in stage environments. DMX512 is a protocol for controlling lighting while MIDI connects electronic instruments together.

By integrating these two interfaces, the Milkymist SoC enables new ways of easily interacting with the visuals.

2.2.11 IR remote

The SoC integrates a RC-5 compatible IR remote control decoder. The user can utilize most electronic appliances' remote controls to interact with the visuals or navigate through GUI menus.

2.2.12 Video input

A video input compatible with PAL, SECAM and NTSC is supported by the system. Like every high-bandwidth DMA master, it is connected to the FML bus.

An external ADC and decoder chip with a BT.656 interface like the ADV7181 is used. This simplifies greatly the problem of decoding multiple video standards.



Fig. 3: Milkymist One (picture: Joachim Steiger, CC-BY-SA)

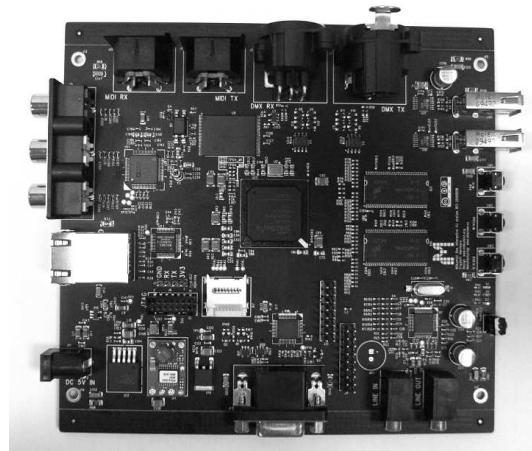


Fig. 4: Milkymist One, case removed (picture: Adam Wang, CC-BY-SA)

This video input enables the use of the device in live video mixing and transformation applications.

3 Hardware development system

The Milkymist SoC is the central component of our commercial Milkymist One product (figures 3 and 4), which uses a cheap and high-density Spartan-6 FPGA (XC6SLX45) to implement it.

Early versions of the SoC were prototyped on a Xilinx ML401 development board equipped with a Virtex-4 XC4VLX25 FPGA.

On the software side:



Fig. 5: The Genode FX GUI toolkit (picture: Genode Labs)

- ISE Webpack from Xilinx synthesizes the FPGA bitstream,
- UrJTAG [10] is used to load bitstreams into the FPGA and write the flash,
- Verilog simulations are run with GPL Cver [15] and Icarus Verilog [17],
- GCC is used to compile the code for the SoC's CPU.

All these tools are either free (as in freedom) or available at no charge.

4 Software support

As stated earlier, GCC can compile code for the LatticeMico32 CPU target, which eases the porting of existing C or C++ software.

Our final system runs RTEMS with kernel drivers for all the system-on-chip components, and uses MTK GUI toolkit (which is a modified version the Genode FX GUI toolkit [11] shown in Figure 5) to enable the user to configure and tune the system. The visual effect engine is based on the “iterative rendering” idea behind MilkDrop, and uses the same system of parametrizable equations to define the effects.

5 Conclusion

Milkymist features a powerful system-on-chip design, perfectly suited for running intensive

video processing and graphics applications. It is also fully open-source (GNU GPL license version 3), flexible and well documented, allowing its components to be re-used in other system-on-chip designs.

Copyright notice

Copyright ©2007-2011 Sébastien Bourdeauducq. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3.

References

- [1] Sébastien Bourdeauducq. *Configuration and Status Register (CSR) bus specifications*, 2009. <http://www.milkymist.org/socdoc/csr.pdf>.
- [2] Sébastien Bourdeauducq. *FastMemoryLink (FML) bus specifications*, 2009. <http://www.milkymist.org/socdoc/fml.pdf>.
- [3] Sébastien Bourdeauducq. *High Performance Dynamic Memory Controller*, 2009. <http://www.milkymist.org/socdoc/hpdmc.pdf>.
- [4] Sébastien Bourdeauducq. *An open hardware VJ platform - Technical aspects*, 2009. <http://www.milkymist.org/socdoc/confslides.pdf>.
- [5] Sébastien Bourdeauducq. *Programmable Floating Point Unit*, 2009. <http://www.milkymist.org/socdoc/pfpu.pdf>.
- [6] Sébastien Bourdeauducq. *Texture Mapping Unit*, 2009. <http://www.milkymist.org/socdoc/tmu.pdf>.
- [7] Milkymist community. *Milkymist - eyecandy on a chip*. <http://www.milkymist.org>.
- [8] Lattice Semiconductor Corporation. *LatticeMico32*. <http://www.latticesemi.com/products/>

intellectualproperty/ipcores/mico32/
index.cfm.

- [9] Silicore Corporation and Open-Cores.org. *WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores*, 2002. http://opencores.org/downloads/wbspec_b3.pdf.
- [10] UrJTAG developers. *UrJTAG*. <http://www.urjtag.org>.
- [11] Norman Feske and Matthias Alles. *Genode FX: an FPGA-based GUI with Bounded Output Latency and Guaranteed Responsiveness to User Input*. <http://www.genode-labs.com/publications/genode-fpga-graphics-2009.pdf>.
- [12] Electronic Engineering Times Via Thomson Dialog NewsEdge. *Lattice spins uClinux support for Mico32*. <http://headsets.tmcnet.com/news/2008/03/31/3357158.htm>.
- [13] Nullsoft. *MilkDrop plug-in for Winamp*. <http://www.nullsoft.com/free/milkdrop/>.
- [14] Tomas Rokicki. *Indexing Memory Banks to Maximize Page Mode Hit Percentage and Minimize Memory Latency*, 2003. <http://www.hpl.hp.com/techreports/96/HPL-96-95R1.html>.
- [15] Pragmatic C Software. *GPL Cver*. <http://gplcver.sourceforge.net/>.
- [16] Shawn Tan. *AEMB 32-bit Microprocessor Core*. <http://www.opencores.org/?do=project&who=aemb>.
- [17] Stephen Williams. *Icarus Verilog*. <http://www.icarus.com/eda/verilog/>.