Bit slicing

Bit slicing is a technique for constructing a processor from modules of processors of smaller bit width, for the purpose of increasing the word length; in theory to make an arbitrary n-bit CPU. Each of these component modules processes one bit field or "slice" of an operand. The grouped processing components would then have the capability to process the chosen full word-length of a particular software design.

Bit slicing more or less died out due to the advent of the microprocessor. Recently it's been used in ALUs for quantum computers, and has been used as a software technique (e.g. in x86 CPUs, for cryptography.[1])

Contents

Operational details

Bit slice processors usually include an arithmetic logic unit (ALU) of 1, 2, 4, 8 or 16 bits and control lines (including carry or overflow signals that are internal to the processor in non-bitsliced CPU designs).

For example, two 4-bit ALU chips could be arranged side by side, with control lines between them, to form an 8-bit ALU (result nee not be power of two, e.g. three 1-bit can make a 3-bit ALU,[2] thus 3-bit (or n-bit) CPU, while 3-bit, or any CPU with higher odd-number of bits, hasn't been manufactured and sold in volume). Four 4-bit ALU chips could be used to build a 16-bit ALU. It would take eight chips to build a 32-bit word ALU. The designer could add as many slices as required to manipulate increasingly longer word lengths.

A microsequencer or control ROM would be used to execute logic to provide data and control signals to regulate function of the component ALUs.

Known bit-slice microprocessor modules:

- 1-bit slice:
  - ...
- 2-bit slice:
  - Intel 3000 family (1974), e.g. Intel 3002 with Intel 3001, second-sourced by Signetics and Intersil[3]
  - Signetics 8X02 family (1977)[4]
- 4-bit slice:
  - National GPC/P / IMP-4 (1973),[5] second-sourced by Rockwell

- National IMP-16 family (1973), e.g. IMP-00A/520D (RALU) with IMP16A/521D and IMP16A/522D, cascadable up to 16 bit
- AMD Am2900 family (1975), e.g. AM2901, AM2903
- Monolithic Memories 5700/6700 family (1974)[6][7][8][9] e.g. MMI 5701 / MMI 6701, second-sourced by ITT Semiconductors
- Texas Instruments SBP0400 (1975), cascadable up to 16 bit
- Texas Instruments SN74181 (1970)
- Texas Instruments SN74S281 with SN74S282
- Texas Instruments SN74S481 with SN74S482 (1976)[10]
- Fairchild 9400 (MACROLOGIC), 4700
- Motorola M10800 family (1979),[11] e.g. MC10800

- 8-bit slice:
  - National IMP-8 family (1974), cascadable up to 32-bit
  - Texas Instruments SN54AS888 / SN74AS888
  - Fairchild 100K
  - ZMD U830C (1978/1981), cascadable up to 32 bit

- 16-bit slice:
  - AMD Am29100 family
  - Synopsys 49C402

Historical necessity

Bit slicing, although not called that at the time, was also used in computers before large scale integrated circuits (LSI, the predecessor to today's VLSI, or very-large-scale integration circuits). The first bit-sliced machine was EDSAC 2, built at the University of Cambridge Mathematical Laboratory in 1956–1958.

Prior to the mid-1970s and late 1980s there was some debate over how much bus width was necessary in a given computer system to make it function. Silicon chip technology and parts were much more expensive than today. Using multiple, simpler, and thus less expensive ALUs was seen as a way to increase computing power in a cost effective manner. While 32-bit architecture microprocessors were being discussed at the time, few were in production.

The UNIVAC 1100 series mainframes (one of the oldest series, originating in the 1950s) has a 36-bit architecture and the 1100/60 introduced in 1979 used nine Motorola MC10800 4-bit ALU[11] chips to implement the needed word width while using modern integrated circuits.[12]

At the time 16-bit processors were common but expensive, and 8-bit processors, such as the Z80, were widely used in the nascent home computer market.

Combining components to produce bit slice products allowed engineers and students to create more powerful and complex computers at a more reasonable cost, using off-the-shelf components that could be custom-configured. The complexities of creating a new computer architecture were greatly reduced when the details of the ALU were already specified (and debugged).

The main advantage was that bit slicing made it economically possible in smaller processors to use bipolar transistors, which switch much faster than NMOS or CMOS transistors. This allowed for much higher clock rates, where speed was needed; for example DSP functions or matrix transformation, or as in the Xerox Alto, the combination of flexibility and speed, before discrete CPUs were able to deliver that.

Modern use

Software use on non-bit-slice hardware

In more recent times, the term bit-slicing was re-coined by Matthew Kwan[13] to refer to the technique of using a general purpose CPU to implement multiple parallel simple virtual machines using general logic instructions to perform Single Instruction Multiple Data (SIMD) operations. This technique is also known as SIMD Within A Register (SWAR).

This was initially in reference to Eli Biham's 1997 paper *A Fast New DES Implementation in Software*,[14] which achieved significant gains in performance of DES by using this method.

Bit-sliced quantum computers

To simplify the circuit structure and reduces the hardware cost of quantum computers (proposed to run the MIPS32 instruction set) a 50 GHz superconducting "4-bit bit-slice arithmetic logic unit (ALU) for 32-bit rapid single-flux-quantum microprocessors was demonstrated."[15]

See also

- Bit-serial architecture

References

1. Benadjila, Ryad; Guo, Jian; Lomné, Victor; Peyrin, Thomas (2014-03-21) [2013-07-15]. "Implementing Lightweight Block Ciphers on x86 Architectures" (https://eprint.iacr.org/2013/445). Cryptology Archive Report 2013/445.
2. "How to Create a 1-bit ALU" (https://web.archive.org/web/20170508194613/https://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Comb/onebitALU.html). *www.cs.umd.edu*. Archived from the original (https://www.cs.umd.edu/class/sum2003/cmsc311/Notes/Comb/onebitALU.html) on 2017-05-08. "[…] Here's how you would put three 1-bit ALU to create a 3-bit ALU […]"
3. "3002 - The CPU Shack Museum" (http://www.cpushack.com/tag/3002/). *Cpushack.com*. Retrieved 2017-11-05.
4. "(unknown)" (ftp://bitsavers.informatik.uni-stuttgart.de/pdf/signetics/_dataBooks/1977_Bipolar_Microprocessor.pdf) (PDF).
5. "IMP-4 - National Semiconductor" (https://en.wikichip.org/wiki/national_semiconductor/imp-4). *En.wikichop.org*. Retrieved 2017-11-05.
6. "6701 - The CPU Shack Museum" (http://www.cpushack.com/tag/6701/). *Cpushack.com*. Retrieved 5 November 2017.
7. "5700/6700 - Monolithic Memories" (https://en.wikichip.org/wiki/monolithic_memories/5700). *En.wikichip.org*. Retrieved 5 November 2017.
8. "File:MMI 5701-6701 MCU (August, 1974).pdf" (https://en.wikichip.org/wiki/File:MMI_5701-6701_MCU_(August,_1974).pdf) (PDF). *En.wikichip.org*. Retrieved 5 November 2017.
9. "Archived copy" (https://web.archive.org/web/20110211171806/http://bitsavers.informatik.uni-stuttgart.de/pdf/mmi/bitslice/6701_4-Bit_Expandable_Bipolar_Microcontroller_Aug74.pdf) (PDF). Archived from the original (ftp://bitsavers.informatik.uni-stuttgart.de/pdf/mmi/bitslice/6701_4-Bit_Expandable_Bipolar_Microcontroller_Aug74.pdf) (PDF) on 2011-02-11. Retrieved 2017-05-21.
10. "SN74S481 - The CPU Shack Museum" (http://www.cpushack.com/tag/sn74s481/). *Cpushack.com*. Retrieved 5 November 2017.
11. Mueller, Dieter (2012). "The MC10800" (http://www.6502.org/users/dieter/a5/a5_6.htm). *6502.org*. Archived (https://web.archive.org/web/20180718220341/http://www.6502.org/users/dieter/a5/a5_6.htm) from the original on 2018-07-18. Retrieved 2017-11-05.
12. "Computers Sperry Univac 1100/60 System" (https://web.archive.org/web/20160611161409/http://bitsavers.trailing-edge.com/pdf/univac/1100/datapro/70C-877-12_8301_UNIVAC_1100_60.pdf) (PDF). Delran, NJ, USA: Datapro Research Corporation. January 1983. 70C-877-12. Archived from the original (http://bitsavers.trailing-edge.com/pdf/univac/1100/datapro/70C-877-12_8301_UNIVAC_1100_60.pdf) (PDF) on 2016-06-11. Retrieved 2016-01-28.
13. "Bitslice DES" (http://www.darkside.com.au/bitslice/). *Darkside.com.au*. Retrieved 2017-11-05.
14. Biham, Eli (1997). "A Fast New DES Implementation in Software" (http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-info.cgi?1997/CS/CS0891). *Cs.technion.ac.il*. Retrieved 2017-11-05.

15. Tang, Guang-Ming; Takata, Kensuke; Tanaka, Masamitsu; Fujimaki, Akira; Takagi, Kazuyoshi; Takagi, Naofumi (January 2016) [2015-12-09]. "4-bit Bit-Slice Arithmetic Logic Unit for 32-bit RSFQ Microprocessors" (http://ieeexplore.ieee.org/document/7350144/?reload=true). *IEEE Transactions on Applied Superconductivity*. **26** (1). doi:10.1109/TASC.2015.2507125 (https://doi.org/10.1109%2FTASC.2015.2507125). 1300106. "[…] 4-bit bit-slice arithmetic logic unit (ALU) for 32-bit rapid single-flux-quantum microprocessors was demonstrated. The proposed ALU covers all of the ALU operations for the MIPS32 instruction set. […] It consists of 348 Josephson junctions with an area of 3.09 × 1.66 mm^2. It achieved the target frequency of 50 GHz and a latency of 524 ps for a 32-bit operation, at the designed DC bias voltage of 2.5 mV […] Another 8-bit parallel ALU has been designed and fabricated with target processing frequency of 30 GHz […] To achieve comparable performance to CMOS parallel microprocessors operating at 2–3 GHz, 4-bit bit-slice processing should be performed with a clock frequency of several tens of gigahertz. Several bit-serial arithmetic circuits have been successfully demonstrated with high-speed clocks of above 50 GHz […]"

External links

- "Untwisted: Bit-sliced TEA time". Archived from the original on 2013-10-21. – a bitslicing primer presenting a pedagogical bitsliced implementation of the Tiny Encryption Algorithm (TEA), a block cipher