مجلة جامعة تشرين للدراسات و البحوث العلمية _ سلسلة العلوم الهندسية المجلد (27) العدد (2) 2005
Tishreen University Journal for Studies and Scientific Research- Engineering Sciences Series Vol. (27) No (2) 2005

Design of Self- Timed Microprocessors

Dr. Radwan Dandeh*

(Accepted 21/2/2005)

□ ABSTRACT □

In the last year virtually all computers design has been on a synchronous logic approach .Now there is renewed interest in asynchronous – or more accurately self-timed, logic based on increasing speed of operation of computer system. A self-timed circuit generates any clock signals may require locally within subsystem. There are many advantages of self-timed logic low power consumption, modularity-data, freedom from clock-skew. Robustnes & typical performance.

***Associate Prof, Department Of System & Computer Networks, Faculty Of Information Engineering, Tishreen University, Lattakia, Syria.**

مجلة جامعة تشرين للدراسات و البحوث العلمية _ سلسلة العلوم الهندسية المجلد (27) العدد (2) 2005
Tishreen University Journal for Studies and Scientific Research- Engineering Sciences Series Vol. (27) No (2) 2005

تصميم معالج صغري متزامن العمل

الدكتور رضوان دندة*

(قبل للنشر في 2005/2/21)

□ الملخّص □

في السنوات الأخيرة معظم الكمبيوترات صممت بالمنطق المتزامن بسبب سهولة التصميم والاختبار. وفي الوقت الحالي، فإن استخدام المنطق الفوري الذاتي المتزامن بحيث إن الساعة يجب أن تصل إلى النظام ككل. في هذه الدارة تولد نبضة ساعة التي يمكن استخدامها داخل أي جزء من النظام. مع العلم أن أجزاء النظام مفصولة عن بعضها. الساعة تستخدم كواجهة (Interface) بين هذه الأجزاء.

تمَّ هنا عمل جديد بحيث يستفاد من هذا النظام في توفير للطاقة بين بوابات النظام، حيث المفاتيح تعمل فقط عند إجراء عملية مفيدة. مع التحرر من نبضات الساعة إذا لم يكن هناك حاجة آنية. بالإضافة إلى إمكانية الاستخدام المتعدد والتطوير الدائم الذي طرأ على تنظيم هذه المجموعات داخل المعالج الصغري.

* أستاذ مساعد في قسم النظم والشبكات الحاسوبية – كلية الهندسة المعلوماتية – جامعة تشرين – اللاذقية – سوريا.

I. Introduction:

In the last twenty years virtually all computer design has been based on a synchronous logic approach. Historically some of the earliest electronic computers used asynchronous techniques but asynchronous logic then lost favour as a design style. There were good reasons for this; synchronous systems have proved more tractable to design and easier to test. Many tools and methods of logic synthesis have been developed as an aid to synchronous logic design. Asynchronous logic has been neglected because it has apparently been more difficult to produce designs than with the 'conventional' approach. However in recent years there has been a growing resurgence in interest in asynchronous digital logic.

One reason for renewed interest in asynchronous or more accurately self-timed - logic is the ever increasing speed of operation of computer systems. In synchronous designs all changes in the state of a system occur simultaneously in response to an externally defined clock signal. This clock must propagate throughout the system which it can only do at a finite speed ultimately limited by the speed of light. This introduces a clock skew as the signal arrives at different points at different times. As the clock frequency increases the skew can grow to be a significant portion of the clock period; this may cause problems and impose a limit to the speed of the system.

A self-timed circuit generates any clock signals it may require locally within each subsystem. Each part of the system is independent of other - physically separated units and there are no clock constraints between subsystems. Instead of using a clock signal to coordinate transfers between functional blocks the transfers are arranged 'privately' between communicating pairs. There is no Problem of clock skew because there is no global clock! The only timing constraints upon the system are those imposed at the interfaces between the subsystems.

2-. Self-Timed vs Clocked Logic

Conventional digital (computer) circuits operate under the control of a central clock: all parts of the circuit change state at the same time under the control-of this clock(fig.1) :

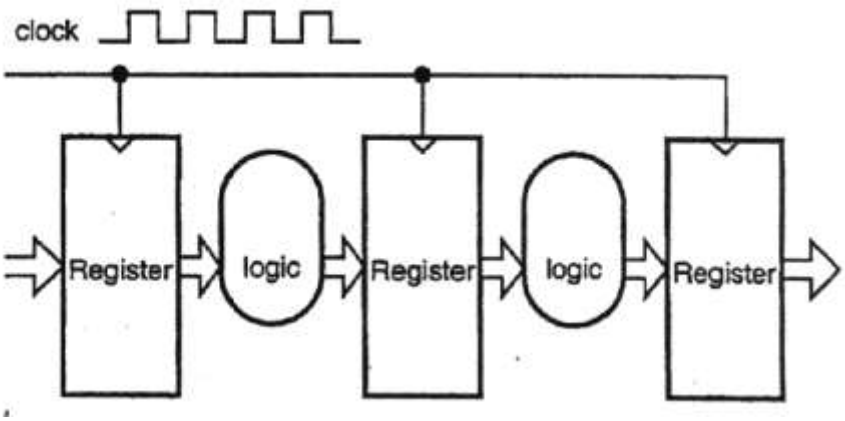

FIG 1:A conventional clocked processing pipeline

Self-timed digital circuits operate without an externally supplied clock to control the timing of data communications. Data is moved between blocks under the control of local Request and Acknowledge handshake signals (fig2):

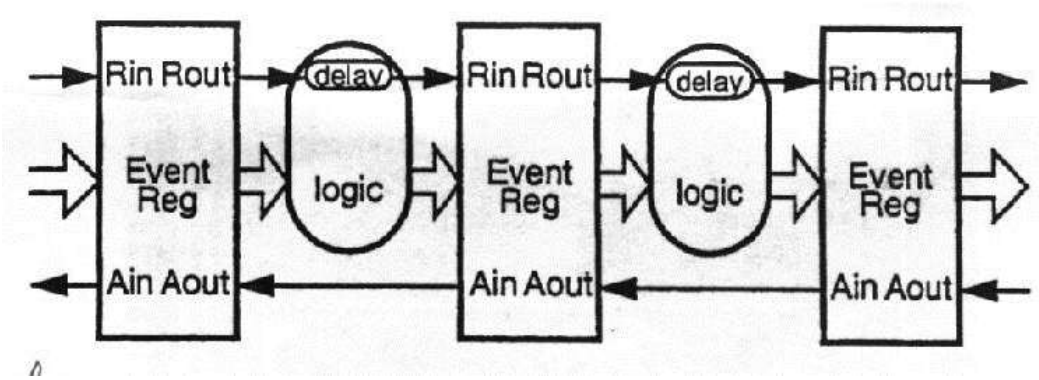

FIG 2: A self-tinted processing pipeline

- The Request signal says when the data is valid;
- The Acknowledge signal says when the data has been received.

In order to maintain the data transfer protocol, the Request line must include a matched delay - a delay which is at least as long as the delay through the processing logic - before it is presented to the following register.

- Unlike a clock, which runs all the time, the Request-Acknowledge handshake is only active when there is data to pass on. This makes selftimed design **low power.**
- Unlike a clocked circuit, where everything switches at the same time, different parts of a self-timed circuit are active at different times. This greatly reduces **radio interference.**

3. Self-Timed Communication:

In a self-timed design there is no clock to control the movement of data; instead, Request and Acknowledge wires are used to implement a full handshake which governs communication. AMULETl uses a 2-phase bundled data convention [1] where valid data is signalled by a transition on the Request wire issued by the sender, and receipt is signalled by a transition issued by the receiver on the Acknowledge wire as shown as fig 3 :

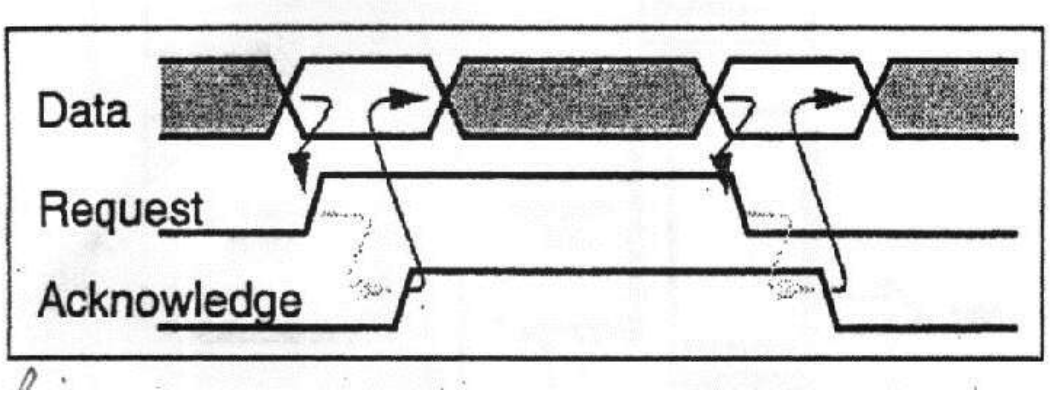

FIG 3 : phase Bundled Data Timing

AMULET2 and 3 use a different signalling scheme where the Request and Acknowledge wires return to zero before the next handshake. This 4-phase bundled data convention is harder to design, but leads to faster and more power-efficient implementations in CMOS.

4. AMULET1

AMULET1 [2-6] was the world's first fully asynchronous implementation of a commercial microprocessor architecture. It demonstrated the feasibility of designing complex chips without clock control.

5. AMULET2

A number of factors combine to make AMULET2 significantly faster and more power-efficient than MlfETl, including the 4-phase control .[7,8] described earlier and architectural improvements such as register forwarding and branch prediction. AMULET2 shows that self-timed designs have competitive performance and advantageous properties in low-power applications.

5.1 AMULET2e

AMULET2e is a system chip comprising an AMULET2 self-timed processor core, a 4 Kbyte associative cache memory and a flexible memory interface. All external accesses are timed using a reference delay attached to the AMULET2e chip, so the board-level design problem is no harder than it is when a clocked processor is used. (fig.4):

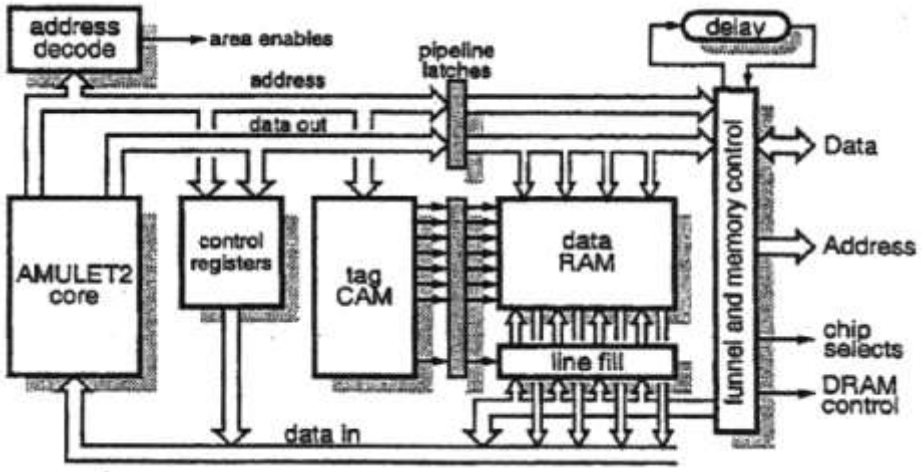

FIG.4: AMULET2eorganisation

6. Amulet3H

FIG.5showsAMULET3H is a self-timed subsystem for use in telecommunications applications. It incorporates a 'bridge' to a synchronous bus which can be used to connect con- ventional clocked telecommunications peripheral con-troljers The AMULET3H subsystem incorporates:

- An AMULET3 self-timed ARM-compatible processor core, including support for "Thumb' compressed instruction set;
  . An 8 Kbyte "local bus' dual-port SRAM memory;
  . A DMA controller;
  . The 'MARBLE, self-timed macrocell bus [11];
  . An interface to on-chip synchronous peripherals;
  . A conventional off-chip memory and peripheral interface.

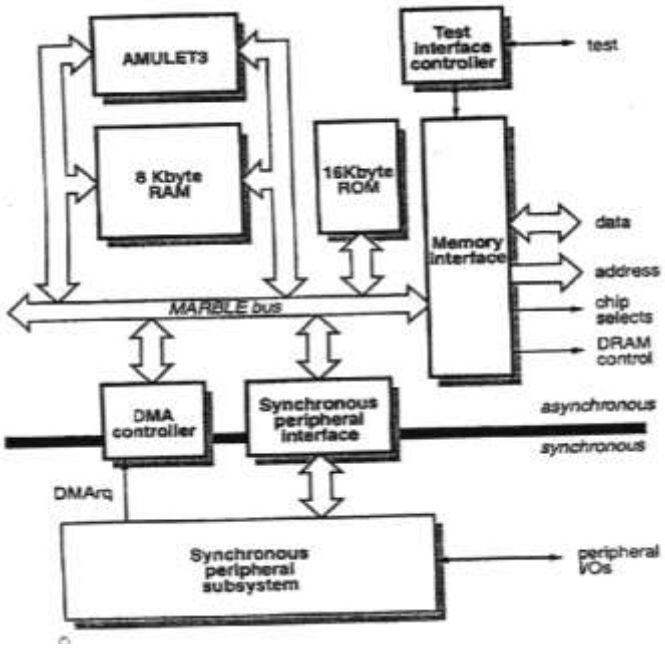

FIG.5: AMULET3H organization

7. RESULTS

. Low power consumption - gates switch only when doing useful work;
. Improved Electromagnetic Compatibility (EMC);
. Modularity - data is encapsulated and there are no global signals;
. Freedom from clock-skew - there is no clock!
. Typical performance - clocks enforce worst-case tolerancing;
. Robustness - the logic speed adjusts to varying environmental conditions.

REFERENCES:

1 Furbe, S-B-, Day, P., Garside, J-D-, Paver, N-C- and Woods, J-V, "A Micropipelined ARM", Proceedings of ' the IFIP TC I0/WG 10.5 International Conference on very Large Scale Integration (VLSI'93), Grenoble France, September 1993. Ed Yanagawa, T. and Ivey, P. A. Pub. North Holland.

2 Furber, S.B., Day, P, Garside, J-D-, Paver, N-C- and Woods, J.v, "AMULETI: A Micropipelined ARM", Proceedings of the IEEE Computer Conference, March 1994.

3 Paver, N-C-, "The Design and Implementation of an Asynchronous Microprocessor", PhD Thesis, University of Manchester 'June 1994'.

4 Furber, S.B., "Computing without Clocks: Micropipeling the ARM Processor", in "Asynchronous Digital Circuit Design", ed. G Birtwistle and A Davis, Proceedings of the l998 VIIth Banff High Order Workshop, Springer, 1995, pp. 211-262.

5 Day, P and Woods, J.v., "Investigation into Micrcpipeline Latch Design Styles", IEEE Trans. on VLSI Systems, June 1995.

6 Furber, S.B. and Day, P, "Four-Phase Micropipeline Latch Control Circuits", IEEE Trans , June 1999, [9] Furber S.B., Garside, J.D., Temple, S., Liu, J., Day P. and Paver, N.C., "AMULET2e: An Asynchronous Embedded Controller", Proc. 3rd Irtl. Symp. on Advanced Research in Asynchronous Circuits and Systems, Eindhoven, April 1997, pp. 290-299. (Best Paper Award)

7 Furber, S.B., Garside, J.D., Temple, S., Day P. and Paver, N.C., "Asynchronous Embedded Control", Journal of Integrated Computer-Aided Engineering, 2000.

8 Bainbridge, W.J. and Furber, S.B., "Asynchronous Macrocell Interconnect using MARBLE", Proc. 4th Intl. Symp. on Advanced Research in Asynchronous Circuits and Systems, San Diego, April 2001 .