ESPRIT 2434

FINAL IMPLEMENTATION REPORT

T I U :

Temporal Inference Unit

Jörg - Ingo Jakob

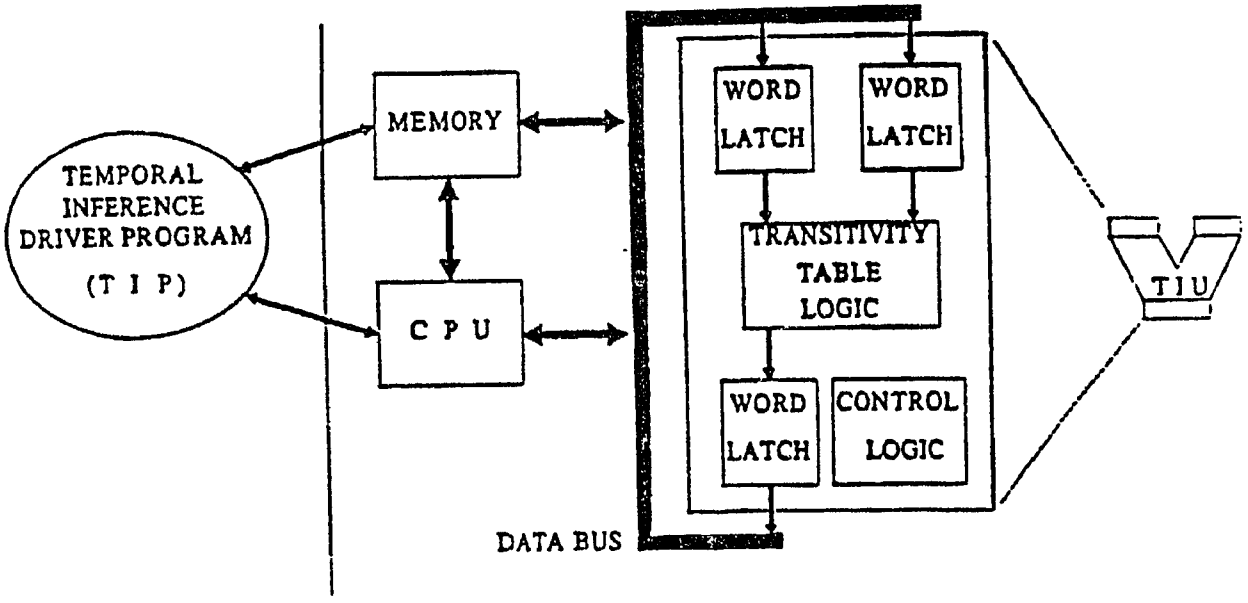
Philips GmbH Forschungslaboratorien
Forschungsabteilung Technische Systeme Hamburg

1. Problem Specification And Motivation

Time logic is used to reason about time. In fact, there are several approaches to time logic. Our application (planning and scheduling in the domain of computer integrated manufacturing) uses an approach known as Allen' qualitative time logic [Allen 1983], enhanced by quantitative extensions. Simple example: "Order 89 is produced on machine M2 before order 133. Order 32 is produced on machine M2 after order 133. What is the time relation between orders 89 and 32 on machine M2?". More formally: "if A before B, and C after B, then holds A before C".

The basic unary (not) and binary (and, or) relations of predicate logic and the inferences arising are built into any conventional CPU, and can be calculated very fast by this CPU. There is no need to introduce a coprocessor for this purpose. This is not true of time logic. The basic time logic relations and inferences arising are not built into a conventional CPU, and it is rather time-consuming to have them calculated by the CPU. There is a practical need to introduce a coprocessor to speed up reasoning about time. This coprocessor we will call a 'Temporal Inference Unit' (TIU).

The TIU is an integrated circuit which we chose to implement as a field-programmable gate array (FPGA). In a minimalistic system, the TIU needs some surrounding hardware to interface with the host system bus. The TIU and its minimal surrounding can be depicted like this:



Temporal Inference Accelerator Unit (TIU)

However, the primary purpose of the TIU is not a standalone coprocessor, but integration into a larger hardware accelarator.

Note: A formal description of this coprocessor is to be found in [Jakob 1990]. A more general introduction is to be found in [Jakob 1991a]. A patent protecting the TIU was recently granted by the German federal patent office.

2. Requirements Specification

A digital hardware device typically meets a "logical" (functional) and a "physical" (electrical) specification. This hardware device shall perform a basic operation of temporal logic in hardware at high speed, as described in Chapter 1 of this product description (Problem Specification and Motivation). The exact representational aspects (data formats) are described in Chapter 6 (Representation and AI Modules). The appendix provides a Technical Data Sheet, concisely specifiying functional. physical and electrical requirements which are to be met by each individual TIU chip which passes fabrication tests.

3. Organization Structure

The design of the TIU is a strategic, low-level part within the activities of subtask 2.5, "System Concept for an Advanced CIM/AI Controller". An overview of the intentions and achievements of this subtask are to be found in the preceding product description of Philips, "TIP: Temporal Inference Propagator and Processor" [Jakob 1991c].

4. Coordination Structure (CIM Controller)

Not applicable to this hardware device.

5. Decision Processes and Decision Nets

Not applicable.

6. Representation and AI Modules

The TIU can be regarded as a mathematical function mapping two bit vectors (words) onto one. A "word" in the context of the TIU is a 13-bit vector. This vector can be mapped onto a list of Allen's 13 primitive relations in a 1 : 1 correspondence: { after [LSB], before, during, contains, overlaps, overlapped-by, meets, met-by, starts, started-by, finishes, finished-by, equals [MSB]} . That is, bit 0 (the LSB = least significant bit) represents the primitive relation 'after', and so on, until bit 12 (the MSB = most significant bit), which represents the primitive relation 'equals'. The presence of a primitive temporal relation in a compound temporal relation is noticed by toggling its corresponding bit to '1', its absence by toggling its corresponding bit to '0'.

Externally, the TIU is one integrated circuit (see Figure A-1 of the Appendix). Internally, the TIU can be decomposed into several functional blocks (Figure A-2): two input latches, one output latch (keeping a word each), combinatorial logic, and a controller circuit.

7. Models

Not applicable.

8. Inference

The TIU is performing the basic (simple-most) inference step in qualitative temporal logic according to Allen [Allen 1983]. In the more general context of constraint satisfaction systems and constraint nets, this inference step is often called "path consistency" (e. g. [Swain and Cooper 1988]).

9. User Interface Guidelines

Not applicable to a hardware device.

10. Hardware And Software Requirements

Being an integrated circuit with a well-defined hardware interface to its (hardware) surrounding, the TIU could be applied in a variety of systems for temporal logic, provided they are able to access a hardware device for speeding up the basic inference step. Typically, the TIU can be used

- as a standalone coprocessor, interfacing to application software by a computer bus (e. g., the PC's ISA/EISA bus); or
- in a larger hardware accelerator for temporal logic, which resides on one or more PCBs (printed circuit boards).

The accompanying product TIP [Jakob 1991c] describes a software and hardware surrounding for the latter application.

11. Prototype Implementation Experiences

Implementing a hardware device typically yields two kinds of experiences: (1) did the device fulfil expectations ?; and: (2) were the available tools suitable for implementing this kind of hardware ?.

The first question can be answered rather easily: the device did fulfil expectations; it works at the required speed (currently: 8 MHz PC/AT ISA bus; planned: EISA bus) and fully conforms to its formal [Jakob 1990] and technical [Jakob 1991b] specification.

Furthermore, the CAE tools available for designing field-programmable gate arrays and the technology of field programmable gate arrays (FPGAs) itself turned out to be mature. At Philips, we used the Actel Logic System (ALS) of the Californian Actel Corporation, which comes with the Viewlogic Workview design and simulation software, capable of producing 8,000 gate FPGA designs on a fast 386 PC. ALS and Actel's FPGA technology have been licensed to several large semicounductor comanpanies all over the world (excluding Europe) during the last two years, amongst them Hewlett-Packard, Texas Instruments and two Japanese corporations[1].

12. Software Module Structure

Not applicable to a hardware device.

13. Software Modules Interface Description

Not applicable to a hardware device.

14. Training Manual

Being a low-level hardware device, there is no training manual available for the TIU. A technical data sheet, parts of which are included in this product description, is to be found in [Jakob 1991b].

15. References

[Allen 1983] J. F. Allen, Maintaining Knowledge about Temporal Intervals; in: Communications of the ACM, Vol. 26 No. 11, November 1983, pp. 832 - 843.
[Jakob 1990] J.-I. Jakob, A Temporal Inference Accelerator Unit; in: ESPRIT 2434 Consortium / Philips GmbH [ed.], ESPRIT 2434 18 Months Report, Supplement B, Task 2.5, Hamburg 1990.
[Jakob 1991a] J.-I. Jakob, Temporal Inference Unit -- Implementing an AI Coprocessor; in: ESPRIT 2434 Consortium / Philips GmbH [ed.], ESPRIT 2434 24 Months Report, Supplement B, Task 2.5, Hamburg 1991.
[Jakob 1991b] J.-I. Jakob, TIU Technical Data Sheet; in: ESPRIT 2434 Consortium / Philips GmbH [ed.], ESPRIT 2434 24 Months Report, Supplement B, Task 2.5, Hamburg 1991.
[Jakob 1991c] J.-I. Jakob, TIP: Temporal Inference Propagator and Processor; in: ESPRIT 2434 Consortium / Philips GmbH [ed.], ESPRIT 2434 Final Report, Part 2, Hamburg 1991.
[Swain and Cooper 1988] M. J. Swain, P. R. Cooper, Parallel Hardware for Constraint Satisfaction; in: Proc. 7th Conf. on AAAI 1988, pp. 682 - 686.

[1] One might argue that Europe again was not alert to a newly emerging technology in the semiconductor business, and did so far not apply for licenses.

Appendix to the ESPRIT 2434 Final Implementation Report on the

Temporal Inference Unit (TIU)

Appendix: TIU Technical Data Sheet (Brief Version)

Index:

Technical Data:

The prototype of the Temporal Inference Unit (TIU) is implemented using Actel's Desktop-programmable Gate Array Technology [Actel]. The TIU is implemented on ACT 1010 gate arrays, which use a 2 µm CMOS programmable antifuse technology. More than 95 % of the 1,200 available gates of an ACT 1010 device are used. The TIU is housed inside an 68 pin PLCC (Plastic Chip Carrier) (figure A-1).

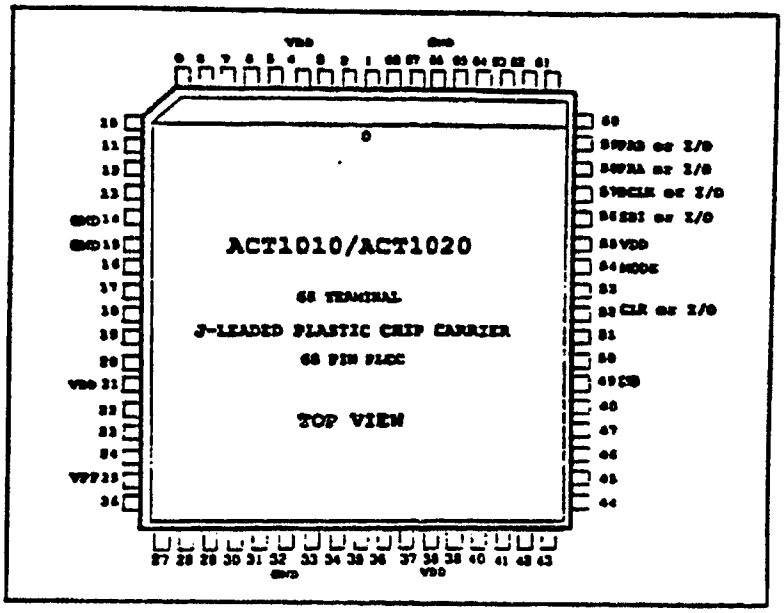


Figure A-1: ACT 1010 device mounted inside an 68 pin PLCC carrier

The internal circuitry of the TIU is described in [Jakob 1990], except for data latches and the necessary control logic to communicate with the TIU's surrounding. The following signals occur (Table A-1):

Table A-1: TIU Signals

Pin	Signal	Signal Description
4	VDD*	VDD = positive supply voltage (+ 5 V)
7	/OE	Output Enable (active low): enable latches of output word
8	EN1	Enable latches of input word 1
9	EN2	Enable latches of input word 2
10	D7	Data[7]
11	D4	Data[4]
12	D1	Data[1]
13	D10	Data[10]
14	GND*	GND = circuit ground (0 V)
15	GND*	
16	D3	Data[3]
17	D6	Data[6]
18	D12	Data[12]

19	D9	Data[9]
20	D0	Data[0]
21	VDD*	
22	D8	Data[8]
23	D11	Data[11]
24	D2	Data[2]
25	VDD*	
26	D5	Data[5]
32	GND*	
38	VDD*	
49	GND*	
54	GND*	
55	VDD*	
56	GND*	
57	GND*	
66	GND*	

Remarks: all starred signals must be connected to the respective potential in order for the TIU to operate properly; none of them may be left unconnected (floating). All pins not in table A-1 must be left unconnected. The 13 bit DATA vector is used for input and output of data words.

A typical TIU is rated to run a TIU cycle (description of TIU cycle: see below) in excess of 10 MHz. When mounted on a ISA bus compatible board, the actual cycle runs at 0.33 MHz due to the speed limitation imposed by the ISA bus.

Functional Description of a TIU Cycle

Definitions:

A "word" in the context of the TIU is a 13-bit vector. This vector can be mapped onto a list of Allen's 13 primitive relations in a 1 : 1 correspondence: { after [LSB], before, during, contains, overlaps, overlapped-by, meets, met-by, starts, started-by, finishes, finished-by, equals [MSB]} . That is, bit 0 (the LSB = least significant bit) represents the primitive relation 'after', and so on, until bit 12 (the MSB = most significant bit), which represents the primitive relation 'equals'. The presence of a primitive temporal relation in a compound temporal relation is noted by toggling its corresponding bit to '1', its absence by toggling its corresponding bit to '0'.

Functional Description:

A functional diagram of the TIU is to be found in figure A-2:

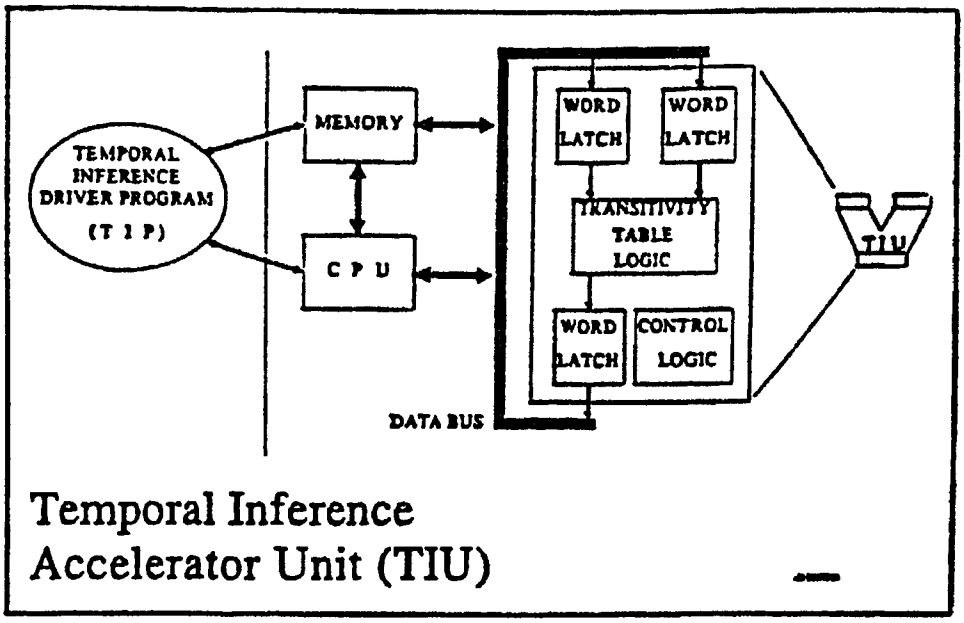


Figure A-2: Functional Diagram of TIU

The TIU always follows its basic processing cycle (unless disabled): A 13-bit vector (input word) is written by the CPU into input word 1. A second 13-bit word is written by the CPU into input word 2. The resulting 13-bit vector appears now at the output latch, it can be read by the CPU.

Word interchange between the CPU and the TIU is done by memory mapped I/O (Intel 386 specific). 13-bit words are expanded to 16 bits and are interchanged by using the 16 bit memory mapped I/O Intel 386 assembler mnemonics (notably, OUT and IN). In our implementation, memory-mapped I/O uses a 6 byte window relative to one user-selectable base address. The bus logic interface boards contains several DIP switches (J1, J2, J3) which can be used to select the base address (standard base address is BA = 300 hexadecimal).

In the beginning, EN1 = 0, EN2 = 0, /OE = 1 (waiting or disabled state). When the CPU addresses base address BA and outputs a 16 bit word by using OUT, the bus interface logic (GAL20V8) sets EN1=1 and thus routes the incoming word to input latch 1. When the CPU addresses BA+2 and outputs another 16 bit word by using OUT, the bus interface logic sets EN2=2 and thus routes the incoming word to input latch 2. Then, the CPU addresses BA+4 and inputs a 16 bit word from the output word latch of the TIU by using IN; the bus interface logic sets /OE=0 and thus routes the outgoing word from the output latch to the CPU.

This cycle continues until the CPU processes other code which does not use the TIU. Then, EN1 = 0, EN2 = 0, /OE = 1 (waiting state) again.

Figure A-3 gives a snapshot of the basic TIU cycle, taken by a logic analyzer:

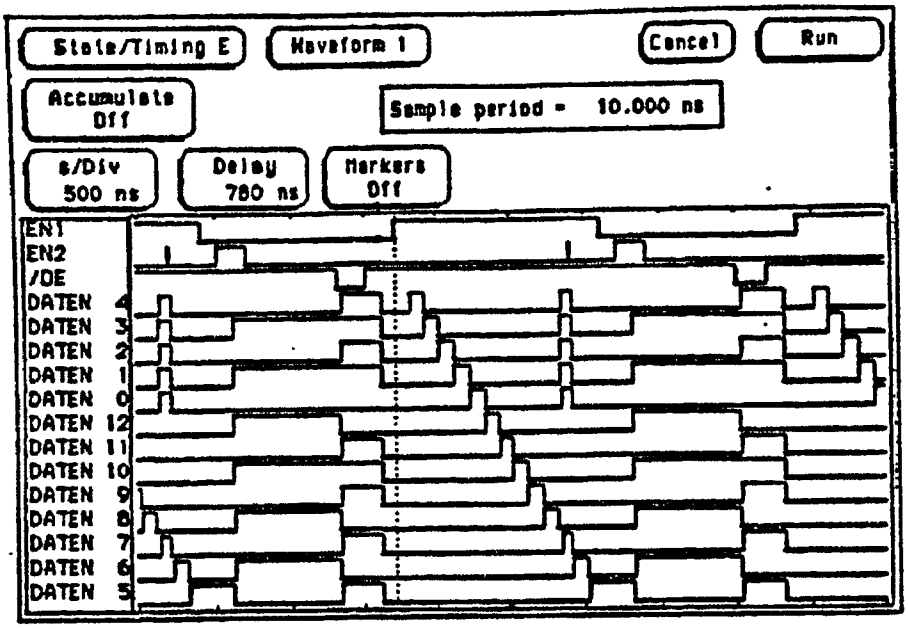


Figure A-3: The Basic TIU Cycle

ISA and EISA Bus Interface Logic

The Bus Interface Logic, which is necessary to enable communication between an ISA or EISA bus host workstation and the TIU, is depicted in figure A-4:

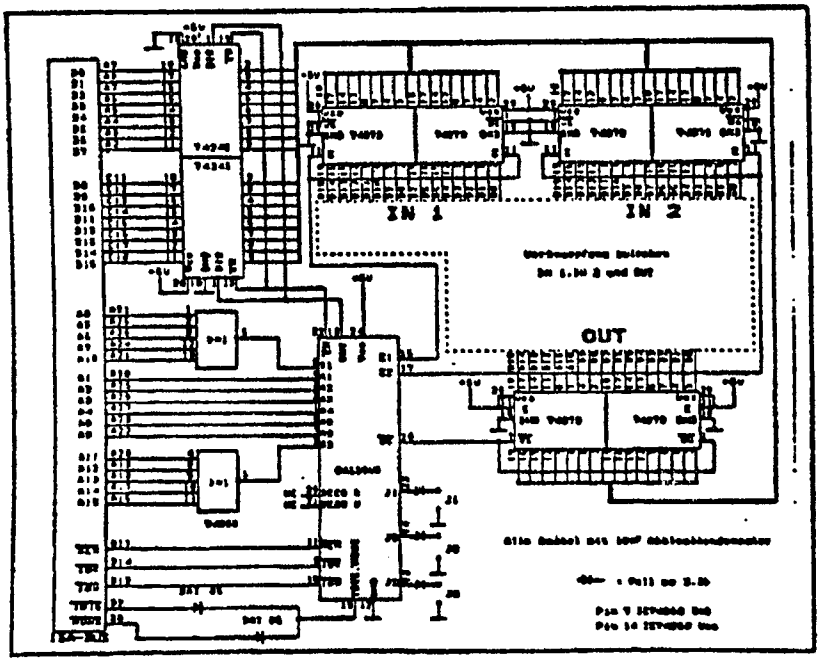


Figure A-4: ISA and EISA Bus Interface Logic

The six 74373 latches are already integrated onto the TIU chip and do not appear on the prototype interface board anymore. They are however retained in figure A-2 to give an expression of the general concept underlying this bus interface board. Any boolean function between two input words and one one output word can be realized using this board. With minor modifications (changing the number of latches), other data word length could be accomodated for. The ISA / EISA Bus Interface Logic as presented here is a general bus interface and not restricted to its use with the TIU.

Figure A-4 gives a schematic of the bus interface logic which connects the TIU with an ISA or EISA bus based workstation. It implements the data exchange between CPU and TIU as descibed above. Note that the 74245 bidirectional buffers are used to route a 16 bit data word either from the bus (CPU) to the TIU, or backwards from the TIU to the bus (CPU). The 74260 NOR gates are used to sense the addresses BA, BA + 2 and BA + 4 from the address bus.