A VMEbus–Interface for the VBN *

A. Gergen, T. Grün, T. Mörsdorf
W. J. Paul, J. Sauermann

University of Saarland
Computer Science Department
Institute for Computer Architecture
Germany, D-6600 Saarbrücken

September 25, 1990

Abstract

The VBN — Vermittelndes Breitband–Netz — of the DBP Telekom is a fast (140 MBit/s) fiberoptical WAN connecting about 300 sites all over Germany. It is known as the 'Videokonferenz–Netz', but it can also do data transmission.

We present an interface card for the VMEbus connecting two *computers directly* via VBN – an alternative to communicate over *networks* (like E-MUX). This card is a D32/D16, A32/A24 VMEbus Slave and an interrupter. It transmits and receives (possibly interrupt controlled) packets of variable length up to 32 KByte at full VBN speed. The bandwidth between computer and interface is about 8 MBytes/s. The full bandwidth of VBN can be exploited by cascading three cards.

1 Introduction

The VBN arose from the 'Video–Konferenz–Netz' of the DBP Telekom. The 'TV-telephone' service was hardly used and hence the network load was very low. So the DBP Telekom decided to establish a further service — high speed data transmission — over the same network under the name VBN. The bandwidth of VBN is about 140 MBit/s; more than 2000 times higher than ISDN with its 64 kBit/s.

The interface at the TAE (Teilnehmeranschlußeinrichtung) of the VBN delivers and requires a ten bit word each 70 ns. For most computers this data rate is too big to deal with directly; one has to attach additional hardware to the computer system.

We chose computer systems with a VMEbus (like e.g. SUN 3). The VMEbus is a fast bus system especially for computers with a Motorola 680x0 CPU. The bus timing is similar to that of the CPU, and DMA (direct memory access) yields data rates up to 40 MByte/s. Data and address busses are both 32 bit wide, but there are also access types using less than 32 bit. VMEbus cards are distinguished in *masters* and *slaves*. Masters generate bus accesses while slaves only reply to them. According to the official VMEbus specification (see [3]) the VBN-VMEbus interface card is a slave (D32/D16, A32/A24, ADO) and an interrupter. The maximum data rate is 8 MByte/s without DMA.

First we explain the mode of operation then we describe the test environment and the software running the interface.

*research partially funded by DFG, SFB 124

2 Mode of Operation

For the design of the interface card [1] a description of the TAE interface [2] was given (see figure 1, right side). There are two ten-bit wide busses — named *SD* and *ED* — with their *clocks*. The signals *state* indicate the current state of the network (ready for sending, ...).
Since the interface shall do communications there is a natural partitioning into transmitter, receiver and VMEbus interface.
Whole packets are transmitted because the protocol overhead for the transmission of single characters would be too high. So we choose the following scheme for the design: The interfaces core is a memory accumulating a stream of characters to a packet. When the packet has been completed it is transmitted undivided over the network.
The transmitter memory is written by the computer and read by the VBN transmission logic — the receiver memory opposite. These memories are special dual ported RAM: writing is associated with the VMEbus and reading is associated with the VBN part. One can realize this type of dual ported RAM as a ping–pong buffer, i.e. memory is split in two banks and the one part is attached to the VBN, the other is attached to the VMEbus. In this way access conflicts are avoided, which can be heavily resolved at that high speed. If the VMEbus bank has been filled with a new packet the banks are switched and the cycle starts again.

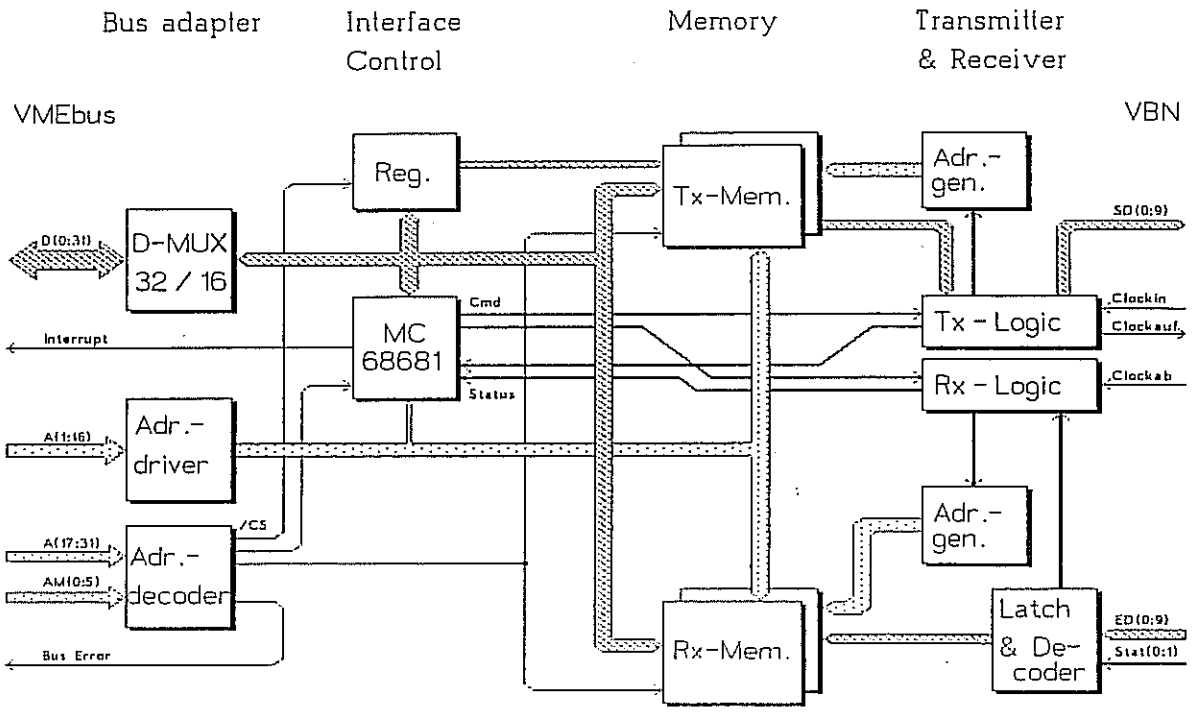

Figure 1: Data paths of the VBN–VMEbus–Interface

The *DUART* MC 68681 controls the interface using its I/O–port. On this card it is rather a multi functional peripherial (MFP) than a DUART. Transmitter and receiver get commands (e.g. 'send packet') from the I/O–port and give back a status word (e.g. 'whole packet transmitted'). At several states the MFP can cause an interrupt. The chip contains a programmable timer too; so software can take advantage of it. The chip does not depend on the special VMEbus timing because its bus interface works asynchronously. This is of interest using other bus systems (MULTIBUS or even SBus).

Both transmitter and receiver consist mainly of an *address generator* generating successive addresses and a *control logic* transferring the data between memory and VBN at the right time. The data sent to the TAE are ten bits wide whereas computers normally use multiples of eight bits. Hence only eight bits are used as real data. In the remaining two bits a channel number is encoded. This number is written once into the *register* and is taken from there each time an access to transmitter memory is made. There are three data channels and one control channel. The control channel carries information about data channels, e.g. 'begin (or end) of a packet in data channel x'.
There are several reasons for such a partitioning: first a packet can have a variable length restricted only by the amount of available memory (here 32 kByte); second one can establish three logical connections over one physical by using three cards and little hardware expansion. So the full bandwidth of VBN can be shared by three computers each bringing only 45 MBit/s data rate.
The interface card is universal though a fixed bus system has been chosen. The DUART and the register do not depend on the special VMEbus timing and the memories are very fast (25 ns). One can modify the interface to use it with other bus systems only by changing *D-MUX* and the *address decoder*.
Big amounts of data can be found in computer graphics, file transfers and LAN–bridging in general. A dedicated fast VMEbus–computer could be installed as remote–server for hard discs or as an ethernet bridge. In this sense the use of VMEbus is no restriction for the user, too.
Normally a point–to–point–connection in the VBN is dialed by hand, but it can also be done by an additional interface called TAV ('Teilnehmeranschlußvermittlung'). The serial line interface necessary for this computer dialing procedure is contained in the MC 68681.

3 Software and testing environment

The actual testing environment consists of two VMEbus–computers with a 68020/68881 CPU under the networking operating system MIRAGE [10]. They are connected by the VBN–VMEbus interface card. One computer is the master of a parallel computer DATIS-32 [4] developped at the University of Saarland. The other serves a high resolution graphics card.
The software submits three different packet types:

- **graphic–packets** are part of the bitmap of an actual picture, which is computed on DATIS-32 and transmitted to the single computers graphics card. Here one can *see* the speed achieved by the connection over the VBN.
- **MIRAGE–packets** are sent by the operating system MIRAGE if it requires data from discs of other computers or if a remote access happened (e.g. terminal emulation). These packets have a fixed length of 1040 Bytes.
- **user–packets** contain data which is exchanged by user processes on two different computers. These packets have a variable length of up to 30 kByte. They bypass the overhead of the operating system MIRAGE. A fast file transfer was realized by this packet type.

A demonstration with fractals was shown at the exhibition CeBIT '90. The pictures were computed in Saarbrücken and transmitted to Hannover via VBN using graphic–packets. The parallel computer DATIS-32 was controlled from the computer in Hannover.
A developed version of the card will be used in the BERKOM project BILUS. There the card will be put in a SUN Sparc Station 1+ managing transfers under TCP/IP.

References

[1] Thomas Grün, 'Entwurf und Realisierung eines Interfaces zwischen dem Vermittelnden Breitband–Netz und Rechnern mit VMEbus', Diplomarbeit, Universität des Saarlandes, 1990

[2] 'Schnittstelle 10 * 13.5 M', Technische Beschreibung, ANT Nachrichtentechnik, 1990

[3] VMEbus Specification, Rev. C.1, Motorola, 1985

[4] W. J. Paul, D. Scheerer, 'The DATIS-P Fault Tolerant Machine', to appear on Hawaii International Conference on System Sciences – 24, 1991

[5] Andreas Gergen, 'Entwurf einer Master–Slave–Kommunikation im Multiprozessorsystem DATIS-P-256', Diplomarbeit, Universität des Saarlandes, 1989

[6] Ina Gläs, 'Die Lösung spezieller Probleme bei der Realisierung eines Permutationsnetzwerks', Diplomarbeit, Universität des Saarlandes, 1989

[7] Thomas Mörsdorf, 'Entwurf, Bau und Test einer 32–Bit–CPU als Teil des Multiprozessorsystems DATIS-P-256', Diplomarbeit, Universität des Saarlandes, 1989

[8] Jürgen Sauermann, 'Ein paralleler APL–Rechner', Dissertation, Universität des Saarlandes, 1990

[9] D. Scheerer, 'Entwurf und Realisierung eines Verbindungsnetzwerkes als Teil des Multiprozessorsystems DATIS-P-256', Diplomarbeit, Universität des Saarlandes, 1989

[10] MIRAGE Assembler Programming System,
MIRAGE Operating System User's Manual, Sahara Software Ltd., 1987