The painful birth of TI99/4

The purpose of this page is to explain the technical choices which were made when designing TI99/4. Incidentally, I tell about a few little-known facts. Information in this page is not guaranteed 100% correct, althought I hope it is. Also, you should understand I am just the compiler and popularizer of information I found elsewhere.

Sources :

- Article by Dan Eicher "The TMS9985", 12/06/1996 (TI Hardware Manual).
- Article by Herbert H. Taylor "TI-99/4 - THE EARLY DAYS", 11/25/1994 (TI Hardware Manual).
- "TI GPL USER'S GUIDE", 12/2/1979.

Quick historic notes

TI99 models :

TI99/4

The original TI99. Released in 1979 (first prototypes in 1978). Was intended to use a TMS9985 microprocessor (or was it a microcontroller ?), but eventually used a 3MHz TMS9900 processor. Features 256 bytes of CPU RAM and 16kb of video RAM.

TI99/7

Never released, developped parallelly with TI99/4, some prototypes built in 1980. Compared with TI99/4, features 64.25kb of CPU RAM, a memory pager, and a better keyboard. It included a speech synthesizer, P-Card, one disk unit, a serial interface, a modem, and an expansion port, all in a single box. Its design seems to prefigure both TI99/8 and TI Business Systems.

TI99/4A

An improved TI99/4. Released in 1981. Features an additional graphics mode, better keyboard, and revised, smaller CPU ROMs.

TI99/8

Never released, some prototypes built in 1983. Features a 10MHz TMS9995 microprocessor, a memory pager, 64.25kb of CPU RAM, improved ROMs, and built-in TI-extended basic, speech synthesizer and P-Card.

TI99/2

Never released, some prototypes built in 1983. Features a 10.7MHz TMS9995 microprocessor, 4.25kb of CPU RAM, and built-in TI-extended basic. To reduce costs, video was simplified, and sound absent. In spite of its name, this computer is not really compatible with TI99/4(A).

TI99/4B & TI99/5

Never released. I only know the name of these machines, even their existence remains uncertain. I guess both names match the same machine, and that it was a TI99/4A with a TMS9995 processor instead of the old TMS9900.

TI99 was abandonned in 1983.

Why TI99 is (almost) 8-bit

TI99/4(A) has frustrating specs :

- Its TMS9900 processor is 16-bit. However, only 8kb of ROM and 256 bytes of RAM are on the 16-bit bus : *everything* else is on an 8-bit-wide, multiplexed bus (even video and sound processor, even thought there was *absolutely* no need for this), which adds 4 wait-state cycles per memory access.
- TI99 uses 3 interrupts. However, they are all mapped to the same vector, hence the interrupt routine has to work out which interrupt was actually triggered. Nevertheless, the TMS9900/TMS9901 couple allows the quite simple use of 15 distinct interrupt vectors !

The reason for all this ? TI99/4 (and /4A) wasn't intended to use a TMS9900, but a TMS9985 microprocessor, with the following specs :

- 8-bit-wide external data bus.
- 256 bytes internal RAM, on a 16-bit-wide bus.
- One single (or two ?) interrupt line associated with one vector (or two ?).

This would have cut costs down. Unfortunately, TMS9985 was never produced industrially (reportedly because of technical problems). As a consequence, a TMS9900 was installed instead. However, as TI was in a hurry, engineers did not change the design at all. They took a TMS9900, added various support device (TMS9901, TMS9904), 256 bytes of RAM, and, to finish this grim work, they put a 16bits → 8bits bus multiplexer and connnected the interrupt pins so that TMS9900 use only one interrupt vector out of 16 available. Later on, they moved the 8kb system ROM from the 8-bit bus to the 16-bit bus.

Note that, as a matter of fact, TI99/8 uses a TMS9995, which is an improved TMS9985.

Why TI99/4(A) is slow

When one watches TMS9900 specs, TI99/4(A), even handicapped by its multiplexer, should be the fastest computer in its generation. But TI engineers did much worse than multiplexing the bus.

Because memory is sluggish

I have already talked about 256 bytes of 16-bit CPU RAM. The question you might raise is : how much 8-bit CPU RAM is there ? Answer : there is none. None in a memory-expansion-less TI99/4(A), at least.

The only available memory, out of the 256 bytes of CPU RAM, is the video processor RAM (16 kb). It has the following characteristics :

- only 8 bits can be accessed at a time (there is even no simulation of 16-bit access) ;
- slow, because it can only be accessed when the VDP is not using it ;
- to access an address which is not right after the last accessed address, we need to reload an address pointer located in the VDP, which takes much time.

But why did TI engineers do such a thing ? After all, the VDP did not need 16 kb of RAM (at least on the TI99/4), so a part of it could have been moved to the CPU bus. The only logical answer is : to cut costs down.

The video processor indeed has the interesting property that it can refresh RAM on its own, which TMS9900 cannot do. (It is necessary to refresh DRAM (dynamic RAM) regularily, otherwise stored data are lost.) If TI had installed some CPU RAM, they should have added a memory refresh logic. They should probably have added an additional address decoding logic, too. Also, it is quite possible that 8 16kbit RAMs (→ 1 * 16 kbytes on an 8-bit-wide bus) were less expensive than, say, 32 4kbit RAMs (→ 4 * 4 kbytes on an 8-bit-wide bus).

You may still wonder why there was no such refreshing issue with the 256 bytes of CPU RAM. The reason is that this RAM is SRAM (static RAM), which does not need to be refreshed. Actually, it was plain silly to install SRAM, because it is very expensive, but keep in mind that TI engineers planned to use a TMS9985 with on-chip RAM, not a TMS9900 with an incredible number of extra chips to simulate a TMS9985.

Indeed, competition had no such problem, since both Z80 and M6502 could refresh RAM on their own. But neither of them was built by TI, so...

Last, note that the 32kb RAM expansion was installed on the CPU bus, and solved all these problems. Thought, most TI99 users did not have one, thus programs which used it were reserved to a "happy few".

Because of GPL

TI99/4(A) massively uses an interpreted language : GPL (Graphics Programming Language). Of course, it is much slower than assembly or any compiled language. Compare, for instance, the execution speed of TI Basic, mostly written in GPL, and TI extended Basic, extensively optimised with assembly.

This has several consequences in the system design. Notably, many functions from the TI99/4(A) operating system are only available in GPL (i.e. unavailable in assembly). Also, on non-paged cartidges, 8kb are available for assembly code, versus 40kb (30kb actually) for GPL code.

There were several reasons for using GPL :

- Written in GPL, a routine is generally smaller than the same routine written in assembly.
- GROMs (ROM with GPL programs) take no CPU address space, which is interesting since TMS9900 can only address 64kb.
- By patenting GPL and by providing informations on GPL only under a non-disclosure agreement, TI could control cartidge production for TI99 completely. Indeed, when ATARI and FUNWARE released independently from TI some cartidges which didn't use GPL, TI *modified the TI99/4A ROMs for them to only accept GPL cartidges*.

Why there are so many weird keyboard modes

In the documentation for the KSCAN system routine (which is used to read the state of keyboard and joysticks), you'll read that several modes can be used. The first one allows to read the keyboard. The 2 next ones allow to read each joystick. Here, it is becoming weird : the joystick modes also read 20 keys from one half of the keyboard (either left or right, according to the joystick). Furthermore, system interfaces are designed to support 7 intensity level in each direction, whereas joysticks sold by TI always had only two states. Had TI engineers planned something else ?

Well, yes, the existence of these modes is not pointless. They should have supported several peripherals which TI abandonned at the last moment.

Here are the peripherals you already know :

- Console keyboard.
- 2 wired joysticks (a.k.a. : remote controllers (name used by TI in its user manuals), wired handsets).

But there should have been a connector (reportedly on the top surface, under the metal overlay) to attach an infrared interface, which would have allowed the use of :

- 2 20-key keypads (handheld unit keyboards, remote handsets).
- 2 additionnal joystick (remote joysticks). These joysticks were "analogic", with three intensity levels in each direction : 7 (full), 4 (medium) and 1 (near).

Before further explanations, note that each I/R joystick was associated with one of the I/R keypad. Similarly, each wired joystick is associated with a half of the console keyboard.

Now, let's explain modes in accordance with TI's original intention (TI99/4 only) :

mode 0

console keyboard.

mode 5

the 2 20-key keypads are mapped together to a 40-key keyboard, equivalent to the console keyboard -- note that TI99/4 only had 40 keys, one of which (space key) is repeated twice (which makes 41 keys).

modes 3 & 4

read an I/R joystick and the associated 20-key keypad.

modes 1 & 2

read a wired joystick, and a half of the console keyboard, which is used as a 20-key keypad.

Key layout and associated system codes for a 20-key keypad

CLR (19)	7 (7)	8 (8)	9 (9)	? (10)
GO (18)	4 (4)	5 (5)	6 (6)	X (11)
SET (17)	1 (1)	2 (2)	3 (3)	NO- (12)
NEXT (16)	STOP (15)	0 (0)	E= (14)	YES+ (13)

How 20-key keypads were simulated on the console keyboard (TI99/4) (author's guess)

Keyboard unit 1					**Keyboard unit 2**				
1 (19)	2 (7)	3 (8)	4 (9)	5 (10)	6 (19)	7 (7)	8 (8)	9 (9)	0 (10)
Q (18)	W (4)	E (5)	R (6)	T (11)	Y (18)	U (4)	I (5)	O (6)	P (11)
SPACE (17)	A (1)	S (2)	D (3)	F (12)	G (17)	H (1)	J (2)	K (3)	L (12)
SHIFT (16)	Z (15)	X (0)	C (14)	V (13)	B (16)	N (15)	M (0)	. (14)	ENTER (13)

		SPACE (unit 1, key 17)			

How 20-key keypads were simulated on the console keyboard (TI99/4A)

Keyboard unit 1						**Keyboard unit 2**					
	1 (19)	2 (7)	3 (8)	4 (9)	5 (10)	6 (19)	7 (7)	8 (8)	9 (9)	0 (10)	=
	Q (18)	W (4)	E (5)	R (6)	T (11)	Y (18)	U (4)	I (5)	O (6)	P (11)	/ (16)
	A (1)	S (2)	D (3)	F (12)	G (17)	H (1)	J (2)	K (3)	L (12)	; (17)	ENTER
SHIFT	Z (15)	X (0)	C (14)	V (13)	B (16)	N (15)	M (0)	, (14)	. (13)	SHIFT	
ALPHA LOCK	CTRL	SPACE								FCTN	

OK, now you know about as much as I do on these keyboard modes. I'd like to know more, particularly on low-level and hardware aspects of the I/R interface. I guess some TI99/4 (all ?) still have the support for those peripherals in ROM, which would be a good source of information. Therefore, if you have a TI99/4 and some competence, have a look at it. (You could look in the KSCAN routine, in the level 1 interrupt routine, and just before it.) If you have a small box with I/R LEDs which connects to your TI99/4, or your TI99/4 seems to have a TMS9985 inside, your fortune is made (maybe) :-) .

If you don't believe me, go and look at the TI diagnostics cartidge (an image is often around on the INTERNET). In the "Handset test", we can see quite clearly *4* joysticks and *4* 20-key keypads. 1979's "TI GPL USER'S GUIDE" also tells about most of what I said, even though it is not as clear.

Why TI99 was a (relative) failure

In my opinion, it was merely doomed from the beginning.

I mean : TI99/4, as TI engineers originally designed it, would have been a primitive, but cheap computer. Instead was released as primitive a computer, which was not so cheap anymore. How could this have worked ?

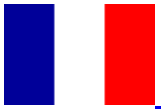
Also, think of the bugs in TI99/4 and its cheap keyboard. It took 2 years for TI to fix them ! (This gave TI99/4A.)

TI99/8 and TI99/2 were more coherent, and might have been more successful (well, TI99/2 sucked, but it would have cost nothing to build). Unfortunately, TI prefered to resign (in 1983). Indeed, it may have been too late. It was just 2 years before the release of ST, and 3 years before the release of Amiga. And the home computer market in 1983 was probably too small for all the many competitors to survive.

TI's commercial policy is another possible cause. I have the impression that they did everything they could to prevent people from writing programs for the computers they sold.

Back to TI99 ressources page.

Back to site index.


French version of this document.

For any comment : pytheas@club-internet.fr