Retrocomputing

Home
Questions
Tags
Users
Companies
Unanswered
TEAMS
Stack Overflow for Teams – Start collaborating and sharing organizational knowledge.
Free!
Create a free Team
Why Teams?

How does the serial port on the RC2014 get addressed?

Asked 2 years, 3 months ago Modified 2 years, 3 months ago Viewed 379 times

Ask Question

I see on the schematic that `/M1` and `A7` go to `CS0` and `CS1` respectively. As I recall, both of these need to be high for the chip to be selected. And `A6` goes to `CS2`, which needs to be low for the chip to be selected.

And of course, `/IORQ` is inverted by the 7404 and fed into `E`, which enables the chip only when the Z80 is accessing the I/O address space.

As I understand, this scheme maps all I/O addresses in the range `0x80` to `0xbf` to the serial port. But what is the involvement of the `/M1` signal here? From what I understand, that basically disables fetching opcodes from the serial port. But the Z80 can't do that anyhow.

I'm asking about this because I'm designing a computer around the RC2014 micro and I want to understand whether I can narrow this range down any. For this reason it would be good to also know which address is actually used by the Microsoft BASIC and Monitor ROMs that ship with the RC2014.

z80 addressing

Share Improve this question Follow

edited Nov 19, 2021 at 8:54

asked Nov 15, 2021 at 20:41
Omar and Lorraine
38.8k ● 14 ● 133 ● 274

@Jumbo please see (and vote on) this answer on meta – Omar and Lorraine Nov 15, 2021 at 20:52

You would probably do better on the RC2014-z80 mailing list via google groups. – Peter! Nov 15, 2021 at 22:25

Well, it really would like to be addressed as 'Sir, could you please send this', but usually ends up being approached as 'Hey you, bang out these bits now' - the harm to its self esteem is simply unknowable... – Jon Custer Nov 16, 2021 at 19:59

Add a comment

2 Answers

Sorted by: Highest score (default)

6

> As I understand, this scheme maps all I/O addresses in the range 0x80 to 0xbf to the serial port.

True

> But what is the involvement of the /M1 signal here? From what I understand, that basically disables fetching opcodes from the serial port.

Again true (if that is somehow made possible at all). But it also protects the ACIA against falsely seeing an interrupt acknowledge as access.

> But the Z80 can't do that anyhow.

Not fetching, but creating a combination of /M1 and /IORQ does happen during interrupts as active M1 and active IORQ signals an interrupt acknowledge cycle. Incorporating /M1 disables that the ACIA could feel accessed by some random value during such.

Insert: Interrupt Acceptance and Acknowledge

Interrupts are only recognized during an M1 cycle when accepted, two states are inserted after T2. The manual calls them wait states, but in reality it's an I/O access marked by /IORQ - except, no I/O address will be outputted. Devices have to detect the combination of /M1 (which is still held active, as we're still within machine cycle M1) and /IORQ as interrupt acknowledge and request for a vector (there is also *no* active /RD present).

This is already mentioned with the signal description on page 7 and 8 of the 1977 technical manual (*1):

$\overline{M1}$ (Machine Cycle one)	Output, active low. $\overline{M1}$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution. Note that during execution of 2-byte op-codes, $\overline{M1}$ is generated as each op code byte is fetched. These two byte op-codes always begin with CBH, DDH, EDH or FDH. $\overline{M1}$ also occurs with $\overline{IORQ}$ to indicate an interrupt acknowledge cycle.
$\overline{IORQ}$ (Input/Output Request)	Tri-state output, active low. The $\overline{IORQ}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{IORQ}$ signal is also generated with an $\overline{M1}$ signal when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus. Interrupt Acknowledge operations occur during M_1 time while I/O operations never occur during M_1 time.

> I'm asking about this because I'm designing a computer around the RC2014 micro and

Cool.

> I want to understand whether I can narrow this range down any.

In that case it's more about the software. As you mentioned,

- the ACIA will answer on any port pair between 080h and 0BEh.
- Is there an 'official' address, like 080h/081h, defined?
- Or are there guidelines suggesting that?
- Does existing software adhere to these guidelines?

After all, compatibility might be only of concern if your intention is to run existing software, written for the RC2014 without any modification. If this is not a goal, you're free to use any other address (in that case I'd suggest to go for full 16 bit decoding).

- Or is the 'knowledge' of this addresses confined to a few drivers or BIOS like routines and not (direct) used by other programs?

This would as well enable a better decoding, as you'd need to only patch/extend these drivers/BIOS routines to comply with your narrowed address range.

> For this reason it would be good to also know which address is actually used by the Microsoft BASIC and Monitor ROMs that ship with the RC2014.

Well, unless there are fixed guidelines, noone may guarantee anything. So, as usual, a grep thru sources is your friend ;))

---

*1 - The first stop, whenever in doubt about some signals, must be the data sheet page describing them. It will usually support previous assumption, or sometimes, really only sometimes, offer a complete new view :))

Next step that would be looking for timing diagrams. They are especially useful when it's about device addressing modes like here (for this case see p.16 of the 1977 technical manual).

Both sections are usually present in all data sheets and quite prominent in manuals.

Share Improve this answer Follow

edited Nov 15, 2021 at 23:34

answered Nov 15, 2021 at 21:04
Raffzahn
220k ● 22 ● 623 ● 910

Add a comment

2

The schematic is slightly confusing as the typical notation for active low signals are left out.

The UART or ACIA is enabled with E high only during IO cycles, so when IORQ is low.

Read/write is defined by RW pin, write being low and read being high.

The UART needs CS0 and CS1 high, and CS2 low to be selected. Which means A7 must be high, and A6 must be low, and M1 must be high.

M1 will be low with IO read active during an interrupt acknowledgement cycle. Thus requiring M1 to be high will ignore interrupt acknowledge cycles and ACIA will only respond to actual IO read cycles.

The IO bus address will be 0b10xxxxxy, which means that the ACIA will respond to all addresses between 0x80 and 0xBF. As address bit A0 is connected to register select pin RS, all even addresses go to control/status registers, and all odd addresses will go to transmit/receive data registers.

As the address decoding ignores 5 bits, the two ACIA registers appear 32 times in the 8-bit address space.

Commonly a program would ever use two addresses at the base of the area to access the chip, so those addresses would be 0x80 for control/status and 0x81 for data.

Share Improve this answer Follow

answered Nov 15, 2021 at 21:23
Justme
30.4k ● 1 ● 68 ● 142

Could you cite any source that it's commonly 80h/81h? – Raffzahn Nov 15, 2021 at 21:27

@Raffzahn You have to understand that about 30 minutes ago I had never heard what an RC2014 is. All variants I found that had an ACIA with same address decoding, so feel free to use any address you want for it, of course, the 0x80/0x81 may not be the only ones people use as long as it works. But for some reason, with this kind of incomplete address decoding, almost always there is some sane logic and simply the base address with unused bits set to zero is used in the code. This applies e.g. to IBM 5150 PC, gaming consoles and many embedded systems. The RC2014 website only mentions 0x80/0x81. – Justme Nov 15, 2021 at 21:40

3 Well, it's just that my experience of 40+ years of programming is that *any* developer left with a choice will, with a high probability, opt to a solution creating maximum compatibility debt. Seems to be a universal law, untouched by sanity. Maybe tied to thermodynamics. But there seems to be hope if, as you state, all documentation is only mentioning 80h/81h. (While I have heard about RC2014, I never looked into at all). – Raffzahn Nov 15, 2021 at 23:33

Add a comment

You must log in to answer this question.

Not the answer you're looking for? Browse other questions tagged z80 addressing .

Linked
5 Is there a way to boot CP/M automatically on a Z80 RC2014?
Related
6 How does the SAM Coupé refresh its RAM?
3 Can't get the AY-3-8913 sound chip to work with my z80 (added schematic)
26 Where does the Z80 processor start executing from?
6 How does the Z80 determine which peripheral sent an interrupt?
4 Z80 Interrupt Mode 2 - How does the CPU know the device address of the interrupting device to execute IN A,(*)?
5 Why is my program outputting only zeros to the data bus during I/O operations?
11 How does Master System backwards compatibility work on the Megadrive?
12 Is I/O port 0xFE reserved on the eZ80?
2 Why does Z80 RC2014 ROM image 88 not have CP/M or BASIC boot options?