



## MVBC INVAP y MET V

23 mensajes

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>  
Para: Martin Ribelotta <martinribelotta@gmail.com>

27 de noviembre de 2019, 16:05

Hola Martin.

Te molesto porque el 9 de enero pasado, hablando de mi proyecto de tesis con el MVBC02D, mencionaste que en INVAP lo usan los atomicos y los de radares MET V. Como estoy con una duda importante sobre si se puede o no rutear directamente (sin intermediario) el bus de datos y el de direcciones entre el MVBC02D y la CPU, tal vez puedas pasarme contacto de gente que haya diseñado hardware con ese micro especializado. Aclaro que estoy viendo el datasheet junto a Pablo Gomez y Diego Essaya (quienes estan en un proyecto de investigacion de UBA sobre la red TCN) y creeme que la hoja de datos es un tanto confusa. Además, encontramos un proyecto OpenSource del grupo que originalmente diseño la norma y se ve claramente en el esquemático que no usan nada entre medio.

Bueno. Perdon por tanto palabrerio.

Saludos.

Gustavo

--

*"Per Aspera ad Astra"*

Gustavo F. Paredes Delaloye

--

[My StartUp](#)  
[My LinkedIn](#)  
[My Torre.bio](#)  
[My Skype](#)

martin ribelotta <martinribelotta@gmail.com>  
Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

27 de noviembre de 2019, 17:12

Hola Gustavo, va entre lineas...

El mié., 27 nov. 2019 a las 16:05, Gustavo F. Paredes - LU2JGP (<lu2jgp@gmail.com>) escribió:

>

> Hola Martin.

>

> Te molesto porque el 9 de enero pasado, hablando de mi proyecto de tesis con el MVBC02D, mencionaste que en INVAP lo usan los atomicos y los de radares MET V. MET II o III no recuerdo... el MET V ya tiene cositas bonitas como buses VME y CANBUS por todos lados...

> Como estoy con una duda importante sobre si se puede o no rutear directamente (sin intermediario) el bus de datos y el de direcciones entre el MVBC02D y la CPU, tal vez puedas pasarme contacto de gente que haya diseñado hardware con ese micro especializado.

Desconozco quien lo diseño... eso fue un diseño legacy de hace muchos años atrás... invap tiene documentación (interna) sobre eso pero creo que incluso los de hardware no recuerdan como o quien lo diseño... mi aproximación a ello fue con la gente de software para hacerles entender como se usaba el asunto del request multibus.

> Aclaro que estoy viendo el datasheet junto a Pablo Gomez y Diego Essaya (quienes estan en un proyecto de investigacion de UBA sobre la red TCN) y creeme que la hoja de datos es un tanto confusa.

Ni lo digas... es un chip viejo pensado para micros viejos (8/16 bits) e incluso para sistemas con microprocesadores tipo 8086 o mc86k

> Además, encontramos un proyecto OpenSource del grupo que originalmente diseño la norma y se ve claramente en el esquemático que no usan nada entre medio.

>

Dependiendo del mecanismo de arbitración al bus puede que no haga falta el control tristate de las lineas del bus... si me pasan el esquemático te digo como lo hacen... Intuyo que pueden estar usando un mc68000 o un 8086/80186 que eran muy comunes en ambientes ruggedized (sin ir mas lejos el MVB del MET II usa un 86k de freescale/motorola)

La mayoría de micros viejos tienen mecanismos multiprocesador (como lineas REQ/GRANT) para advertirle que tiene que abandonar el bus por los siguientes ciclos para que algun coprocesador como este (o en los microordenadores viejos el controlador de video o dma) accedan a la memoria.

Por desgracia, el diseño de los buses para los procesadores actuales inferen que no hay nada compartiendo el bus, mas que nada por una cuestion de ruteo y velocidad. Sino no seras capas de llegar a un bus de 50-80MHz teniendo un ruteo multipunto como los antiguos buses de procesadores...

El esquema generico de coneccionado al bus es este:

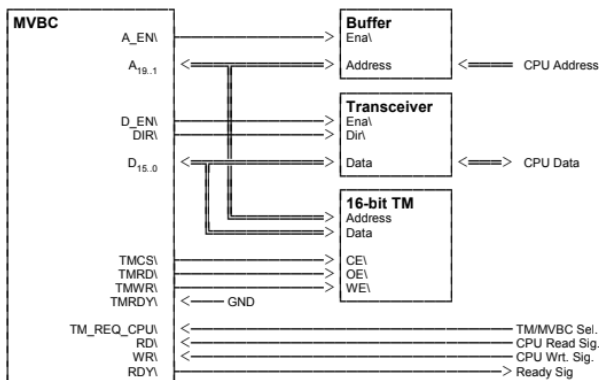


Figure 1.4: Simple CPU / MVBC / TM Interconnection Diagram

Donde se puede ver perfectamente como los buffers tristate (unidireccionales en el caso de address y bidireccionales en el caso de datos) aíslan una posible CPU sin posibilidad de poner sus buses en modo tristate del ciclo de acceso a memoria del MVBC.

Normalmente el MVBC puede acceder a la memoria sin problemas ya que los buffers/trainseiver están en modo de alta impedancia del lado del bus de la TM y por mas que los mueva el CPU principal no va a haber cortocircuito nunca.

Cuando el CPU requiere un acceso a la TM, pone a bajo la linea TM\_REQ\_CPU, el MVBC termina la operacion que estaba en curso (si la habia), habilita los buses de acceso y pone en bajo la linea "RDY" para avisarle a la CPU que el bus es accesible.

Mientras "RDY" este en alto, la CPU principal debe abstenerse de acceder al bus porque los buffers/tranceivers están en modo de alta impedancia y cualquier movimiento de las líneas no accederán a la memoria TM.

Solo cuando "RDY" esta bajo, el CPU principal esta seguro de que el acceso al bus esta ocurriendo de forma correcta ya que el chip MVBC tiene A\_EN, D\_EN y DIR en los valores correctos para la operación (fijarse que DIR es la mas compleja ya que es una conjunción del RD y WR que vienen de la CPU). Eso sin contar que también maneja las señales TMRD/TMWR/TMCS que, en caso de estar liberado el bus de la TM estas son un calco de las señales RD/WR/TM\_REQ\_CPU.

También entiendo que no solo puedes acceder a la memoria TM desde el bus de datos sino que el MVBC entiende determinado rango de direcciones como de registros internos y eso hace aun mas complejo el manejo que tiene que hacer este chip de las señales TMRD/TMWR/TMCS (básicamente, si a0..19 están dentro del rango de los registros, no activa TMCS y accede a los registros internos comportándose como una memoria de cara a la CPU)

Como vez, el acceso al bus en multiprocesador es extremadamente delicado y requiere de bastante arbitración y mucho cuidado de como se sincronizan las cosas para no generar un daño a la circuiteria.

Del otro lado, en la siguiente figura, para no requerir bus tranceiver, se presume que la CPU tiene capacidad de GRANT/REQ/BUSY (req avisa que se requiere el bus, grant señala que el controlador lo concedió y BUSY señala que el master actual del bus sigue ocupado)

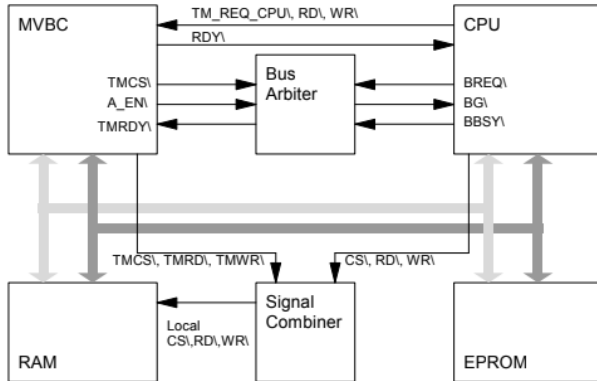


Figure 3.11: DMA Direct Mode Principle

Eso, sumándole a que hay una lógica de arbitración del bus con cierta inteligencia para prevenir las colisiones de acceso.

La única cosa de la que disponen los buses de los procesadores actuales, es de un conecionado punto a punto entre la CPU y la memoria y, a lo sumo, una señal de WAIT para retrasar la lectura del dato hasta que la memoria tenga los datos validos (normalmente eso viene de memorias como las flash lentas que la generan en algunos casos.... nada que sirva realmente aquí)

Estrictamente hablando la señal "RDY" del controlador podría oficiar como señal de WAIT ya que retrasa la lectura del bus, pero esto no implica que el CPU este abandonando el bus y poniendo sus pines en alta impedancia... (de hecho mantiene address, data y control sin cambios a la espera de que la supuesta memoria reaccione) cosa que no ocurre a menos que se fuerce por software (cambiando la configuración de los pines en un proceso lento y molesto de hacer muchas veces)

En primera instancia yo pensaría que el mecanismo de GRANT/REQ se puede emular con software, es decir, conectando la señal "RDY" del MVBC a una interrupción de la CPU y cambiando la configuración de los pines de address/data desde "Analog" (o sea, no conectados a nada) a ALTFUNC (conectados al controlador de memoria) para luego operar, retornar al estado "Analog" y liberar la señal TM\_REQ\_CPU...

Eso ultimo podría funcionar en un ambiente de laboratorio, pero cualquier error de software podría generar daños físico a los componentes así que yo recomiendo fuertemente que uses bus transeivers que son baratos, comunes y hasta te ayudarían al ruteo ;-)

Por lo que entiendo, para el address necesitas un buffer tristate de 18 bits (a1...a19) o un par de 9 bits (son comunes por el tema de la paridad pero pueden ser caros....) yo los emularia con dos chips con 8 buffers tristate normales y dos buffers tristate single de esos chiquititos que vienen en SOT-23

Por el lado del address se complica un poco mas porque tenes que usar un transeiver con control de direcciones... por suerte eso esta resuelto hace 40 años con la misma familia de chips que se basan en el 74LS245... a continuación un producto de TI que es un calco del viejo:

<http://www.ti.com/lit/ds/symlink/sn54ls245-sp.pdf>

Seguro que hay buffers bidireccionales de 16 bits por ahí... pero todo eso te aumentaría los costos y es preferible meter guita en que los componentes sean realmente de grado industrial/automotive a que sean super funcionales...

Por supuesto, tenes que tener cuidado con el 74LS245 de no pifiarle al lado A y B para que DIR te quede en la polaridad correcta y no te pase como a mí en un diseño de hace muchos años con un Z80 que termine haciendo cadorcha por eso...

> Bueno. Perdon por tanto palabrerio.

>

Touche... yo redoblo el palabrerio jajajaja

[El texto citado está oculto]

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>

27 de noviembre de 2019, 17:20

Para: martin ribelotta <martinribelotta@gmail.com>

Gracias.

Si. Eso de la arbitración me daba pie para hacerlo directo, pero bueno. Ahora que lo decis vos, agarre mas confianza.

Si. Habia visto los 74LS245 en Digikey.

Creo que voy por la coneccion directa con un buen firmware.

Te adjunto el esquematico OpenSource.

Gustavo

[El texto citado está oculto]



kp99s\_schema.pdf

528K

martin ribelotta <martinribelotta@gmail.com>

27 de noviembre de 2019, 17:23

Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

Jajajaja ahora entiendo claro... con un FPGA todos somos mandrake jejeje...

Puede que para hacer un prototipo te sirva pero si pensas en la seguridad funcional, me daría cosa que el software tenga la posibilidad de quemar el hardware tan fácilmente...

Bueno, suerte con eso y cualquier cosa acá estoy...

[El texto citado está oculto]

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>

27 de noviembre de 2019, 17:48

Para: martin ribelotta <martinribelotta@gmail.com>

Jajaa... Me dejás pensando...

Tenes razon. No es un buen diseño si le resulta tan facil al soft romper todo...

Lo que me llama la atención del diseño OpenSource es que los buses estan unidos fuera de la FPGA. O sea, no podes implementar una buffer ni nada por el estilo en esa FPGA porque todo llega unido a los pines de la FPGA.

Bien dice el proyecto que es un diseño para empezar (o algo asi).

Gustavo

[El texto citado está oculto]

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>

27 de noviembre de 2019, 17:51

Para: Pablo Gomez <elpablogomez@gmail.com>, Diego Essaya <dessaya@gmail.com>

Buenas tardes.

Me decidi.

Si leen mas abajo las respuestas de Martin Ribelotta, no hay duda que un buen diseño debe incluir buffers. Caso contrario, el soft puede romper el hard con una linea de codigo.

Asi que si no hay objeciones, le inserto buffers entre medio para que todo sea de 10.

Obviamente que puede funcionar sin los buffers, pero....

Saludos.

Gustavo

[El texto citado está oculto]

**martin ribelotta** <martinribelotta@gmail.com>

Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

El mié., 27 nov. 2019 a las 17:48, Gustavo F. Paredes - LU2JGP (<lu2jgp@gmail.com>) escribió:

Jajaa... Me dejás pensando...

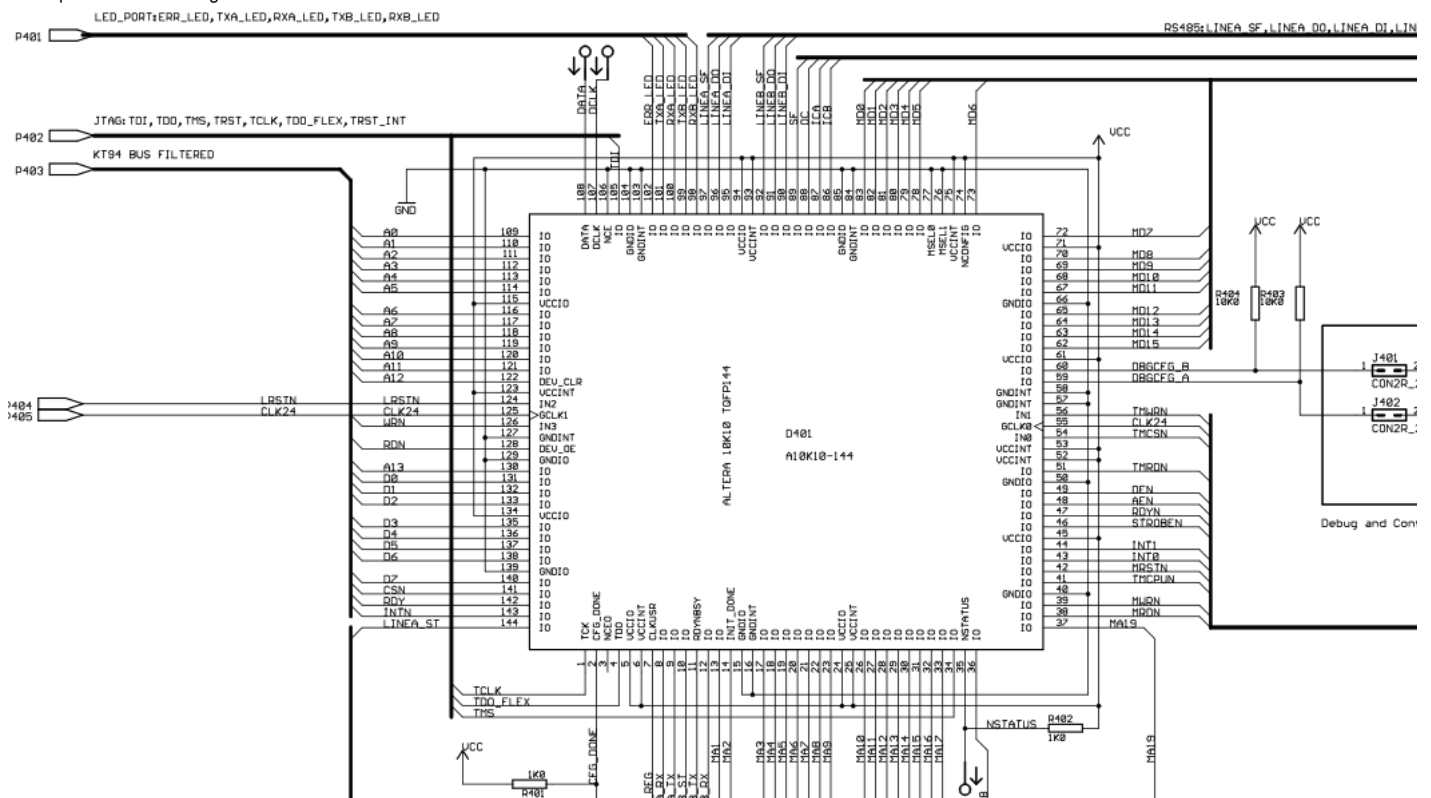
Tenes razon. No es un buen diseño si le resulta tan facil al soft romper todo...

De por si el ruteo es bastante molesto y vas a tener que usar por lo menos 4 layers (vcc/gnd internos y top/bottom como señales) posiblemente usando todo el bottom para rutear las señales del pinout del micro hacia los transceivers y de ahí a la memoria/MVBC que estan mas ordenados... como yo lo veo, separar con buffers te permite ordenar las señales y mejorar el ruteo organización...

Otra cosa ahora que me acuerdo, usa resistencias terminadores en serie con las lineas de datos y direcciones... en principio serian de 0ohm pero si vez rebotes en vez de bajar la velocidad de resistencia para que estas se coman los rebotes y mejore un poco la señal... igual estos son dispositivos lentos (por debajo de los 10MHz estimo) así que no deberias tener problemas de señal...

Lo que me llama la atención del diseño OpenSource es que los buses estan unidos fuera de la FPGA. O sea, no podes implementar una buffer ni nada por el estilo en esa FPGA por la FPGA.

De lo que entiendo del diagrama:



Es que vos tenes dos buses... el bus A0...13 y D0..7 (bus de 8 bits, posiblemente paginado de alguna manera) y el del chip controlador MA0...19 MD0..16. Entonces los buffers y demás los implementas internamente en el FPGA con multiplexores... Desde mi punto de vista, ya que metieron un FPGA podrian haberse obviado el chip de programación.

[El texto citado está oculto]

**Diego Essaya** <dessaya@gmail.com>

27 de noviembre de 2019, 18:14

Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

Cc: Pablo Gomez <elpablogomez@gmail.com>

Qué grande Martín, esa explicación que se mandó vale oro.

De acuerdo en que es mejor ir por lo seguro y agregar esos chips para que no quede posibilidad de dañar el mecanismo eléctrico del MVBC.

Sabés qué chips usar? Se consiguen fácil?

[El texto citado está oculto]

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>

Para: martin ribelotta <martinribelotta@gmail.com>

27 de noviembre de 2019, 18:15

Si, todos los otros diseños que averigüe usan FPGA en lugar del Bombardier (para no depender de ese proveedor, decían).

Pero si miras la hoja 5 vas a ver que el bus de datos y el de direcciones estan unidos. No me refiero a unidos entre si (datos y direcciones) sino a que hay bifurcaciones en cada uno de ellos hacia la memoria (obviamente), el MVBC01 (en este caso, pero es lo mismo que el 02D) y hacia la FPGA.

Para implementar dentro de la FPGA algo que divida, deberian llegar por un pinout y salir por otro. Solo asi podes poner algo "en el medio". Como esta hecho, la FPGA debe respetar las reglas de arbitración que detalla la hoja de datos del Bombardier y que vos me detallaste rebien.

En resumen, no respetaron lo que recomienda la hoja de datos ni aun usando una FPGA que digamos... te permite ser Gardel....

Asi lo veo yo.

Gustavo

[El texto citado está oculto]

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>

Para: Diego Essaya <dessaya@gmail.com>

Cc: Pablo Gomez <elpablogomez@gmail.com>

27 de noviembre de 2019, 18:17

El me tira un dato de uno. Yo anduve viendo algo el otro día. Pero el me diferencia entre manejo de datos y manejo de direcciones. Yo no entiendo la diferencia en usar chips de modelos diferentes para cada caso. Obviamente si debe haber un buffer para el bus de direcciones y otro para el de datos.

Sin desmerecer, pero es lo que dice el datasheet en el punto de arbitracion. Igual. Martin lo sabe de memoria...

Gustavo

[El texto citado está oculto]

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>

Para: martin ribelotta <martinribelotta@gmail.com>

27 de noviembre de 2019, 21:41

Bueno.  
Busque y busque y filtre y filtre en DigiKey.

Para el address elijo el SN54LVTH18514

Para el data elijo el SN74ALVCH16334

Van adjuntos.

Que opinas ??

Saludos.



Gustavo

El mié., 27 nov. 2019 a las 17:12, martin ribelotta (<martinribelotta@gmail.com>) escribió:

[El texto citado está oculto]

[El texto citado está oculto]

2 adjuntos

-  **sn74alvch16334.pdf**  
159K
-  **sn74lvth18514.pdf**  
580K

Gustavo F. Paredes - LU2JGP <lu2jgp@gmail.com>

Para: Diego Essaya <dessaya@gmail.com>

Cc: Pablo Gomez <elpablogomez@gmail.com>

27 de noviembre de 2019, 21:42

Bueno.  
Busque y busque y filtre y filtre en DigiKey.

Para el address elijo el SN54LVTH18514

Para el data elijo el SN74ALVCH16334

Van adjuntos.



Que opinan ??

Saludos.

Gustavo

[El texto citado está oculto]

2 adjuntos

-  **sn74alvch16334.pdf**  
159K
-  **sn74lvth18514.pdf**  
580K

martin ribelotta <martinribelotta@gmail.com>

Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

27 de noviembre de 2019, 22:50

El mié., 27 nov. 2019 a las 21:42, Gustavo F. Paredes - LU2JGP (<lu2jgp@gmail.com>) escribió:

Bueno.  
Busque y busque y filtre y filtre en DigiKey.

Para el address elijo el SN54LVTH18514

Para el data elijo el SN74ALVCH16334

Van adjuntos.

Que opinas ??

No sería al revés? El de 20 bits es unidireccional (address A1...A19) y el de 16 bits sería el bidireccional?  
Si se entendió al revés de lo que dije me disculpo...

Acá te hice algunos filtros:

Buffer tristate bidireccionales (para data) de 16 bits (algunos son dos de 8 en el mismo encapsulado): <https://www.digikey.com/short/pvbt3j>  
Buffer tristate unidireccionales (para address) de 18 bits (algunos son dos de 8 en el mismo encapsulado): <https://www.digikey.com/short/pvbt3r>

Los mas polenta son los de texas instrument pero nexperia es un jugador tan bueno como TI (tiene atras las foundries de NXP y su tecnologia es de philips)  
[El texto citado está oculto]

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>  
Para: martin ribelotta <martinribelotta@gmail.com>

27 de noviembre de 2019, 23:10

Me puedo haber confundido yo (casi seguro).

Gracias por las listas y la data de nexperia.

Mañana las pulo.

Gustavo  
[El texto citado está oculto]

**Diego Essaya** <dessaya@gmail.com>  
Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>  
Cc: Pablo Gomez <elpablogomez@gmail.com>

28 de noviembre de 2019, 0:06

Veo que los que elegiste tienen una señal de clock... es necesario?

Lo que entiendo del diagrama del MVBC es que no... así que me puse a buscar algunos con solo la señal de Ena / OE (output enable) como se ve en el diagrama.

Qué tal este buffer? <http://www.ti.com/lit/ds/symlink/sn74lvc16244a.pdf> (elegí la versión LV que funciona hasta 3.6v, no sé si es la apropiada pero hay otras variantes para elegir).  
Acá hay uno de 32 bits: <http://www.ti.com/lit/ds/symlink/sn74lvth32244.pdf>

Y un transceiver de 16 bits <http://www.ti.com/lit/ds/symlink/sn74lvc162245.pdf>

[El texto citado está oculto]

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>  
Para: Diego Essaya <dessaya@gmail.com>  
Cc: Pablo Gomez <elpablogomez@gmail.com>

28 de noviembre de 2019, 8:33

Hice una eleccion mas simple. Solo mire tensiones y cantidad de bits. Justo en otro email que no salio te decia que mañana (por hoy) purgaba mas esa lista de posibles chips. Me ganaste de mano... Jejeee...

Por lo que veo ahora rapidito, estan bien. Despues miro con mas detalles.

El clock no va. Es todo asincronico. La memoria tambien. Aunque en el punto 9.1 de la pagina 126 indica que se puede usar sincronico tambien.  
Por eso hay señalamientos que indican cuando se puede usar el bus y cuando no. Si bien es configurable el tema de la arbitración (quien tiene mas prioridad), yo pienso dejarlo en el que da mas prioridad al MVBC02D sobre la CPU (ARB = 1). Ver punto 3.12 pagina 73.

Saludos.

Gustavo  
[El texto citado está oculto]

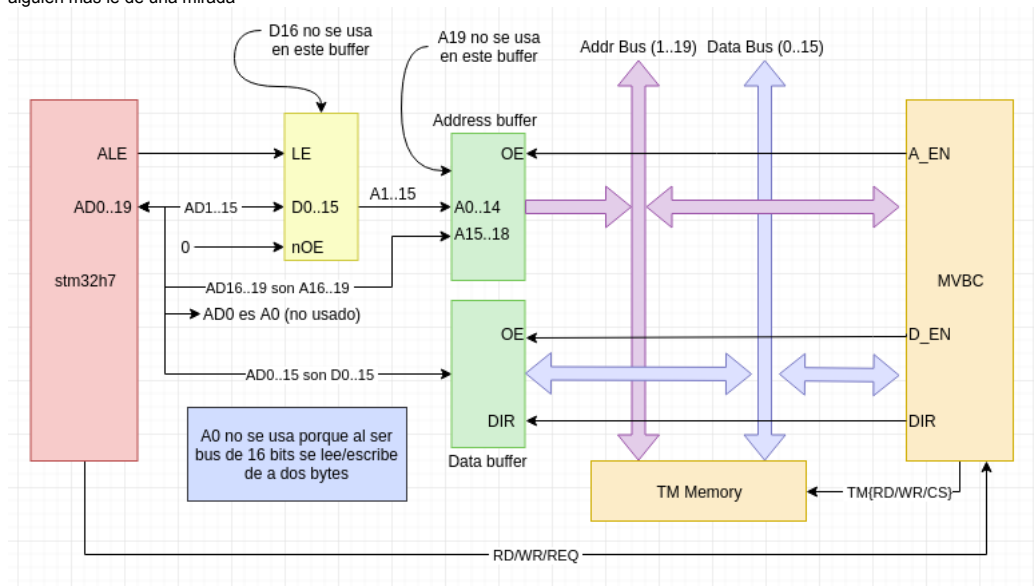
**martin ribelotta** <martinribelotta@gmail.com>  
Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

28 de noviembre de 2019, 20:31

Estuve mirando un toque lo que te mande y el buffer de address tiene que bancarse 19 bits al menos (A1...19) y el que te mande es de 18 jejeje  
Acá hay uno de texas que es de 20 bits (la entrada que sobra mandala a masa no hay drama)  
<https://www.digikey.com/product-detail/en/texas-instruments/SN74AVC16827DGV/296-9801-1-ND/380330>

Pero es sustancialmente mas caro... una cosa que se podría hacer sin aumentar tanto el costo es usar uno de estos buffers tristate chiquitos:  
<https://www.digikey.com/product-detail/en/texas-instruments/SN74AUP1G240DBVR/296-17917-1-ND/758778>

Ademas recuerda que necesitas un latch para transformar AD1..15 en A1...15 junto con la señal ALE... todo creo que quedaria asi... por las dudas revisalo vos mismo y fijate que alguien mas le de una mirada



[El texto citado está oculto]

---

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>  
Para: martin ribelotta <martinribelotta@gmail.com>

4 de diciembre de 2019, 16:47

Genial. Gracias.  
Recien hoy pude sentarme a mirar de nuevo esto...

Gustavo  
[El texto citado está oculto]

---

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>  
Para: martin ribelotta <martinribelotta@gmail.com>

4 de diciembre de 2019, 16:58

Otra pregunta. Con que haces esos graficos ?. Veo que tenes una facilidad para hacerlos... Aunque por tu trabajo, supongo que muchas veces tenes que hacer esos diagramas para tus clientes.

Gustavo  
[El texto citado está oculto]

---

**martin ribelotta** <martinribelotta@gmail.com>  
Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>

4 de diciembre de 2019, 17:00

Uso [draw.io](https://www.draw.io) que es una app web... la podes usar online (util si no estas en tu maquina) o desde una app que te bajas basada en electron...

<https://www.draw.io/>

Te deja usar google drives y otros para guardar cosas o bajartelo a la maquina...  
[El texto citado está oculto]

---

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>  
Para: Pablo Gomez <elpablogomez@gmail.com>, Diego Essaya <dessaya@gmail.com>

4 de diciembre de 2019, 17:03

Hola.  
Reenvio email de Pablo Ridolfi de hace una semana (sip. Pasan rapido los 7 dias...).  
Me hizo un detallado grafico (envidiable este pibe como labura) y en base a eso (que era lo que ya veniamos hablando (o cerca)) elegi los siguientes buffers.  
Datos: sn74lvcr162245 El transceiver que Diego me aconsejo en su ultimo email.  
Address: sn74avc16827 El que me recomendo Martin. Dice que el precio es caro, pero 4 dolares no es tanto y todo esta en un solo chip.

Bueno. Me pongo a terminar el esquemático.

Ah... Antes tengo que hacer el informe de avance. Ariel nos sentencio hasta este viernes !!!

Saludos.

Gustavo

----- Forwarded message -----  
De: **martin ribelotta** <martinribelotta@gmail.com>  
Date: jue., 28 nov. 2019 a las 20:31  
Subject: Re: MVBC INVAP y MET V  
[El texto citado está oculto]  
[El texto citado está oculto]  
[El texto citado está oculto]

---

**Gustavo F. Paredes - LU2JGP** <lu2jgp@gmail.com>  
Para: martin ribelotta <martinribelotta@gmail.com>

4 de diciembre de 2019, 17:04

Ahhhh,, si si. Ya lo use. Genial !!!

Gustavo  
[El texto citado está oculto]

---

**Pablo Gomez** <elpablogomez@gmail.com>  
Para: "Gustavo F. Paredes - LU2JGP" <lu2jgp@gmail.com>  
Cc: Diego Essaya <dessaya@gmail.com>

5 de diciembre de 2019, 8:43

Impresionante! Pusiste Ridolfi pero entiendo te referís a Martín, ¿no? Por favor no dejes de agradecerle por su colaboración.

Estoy de acuerdo y no hay problema con el valor del integrado.

Saludos!  
Pablo  
[El texto citado está oculto]