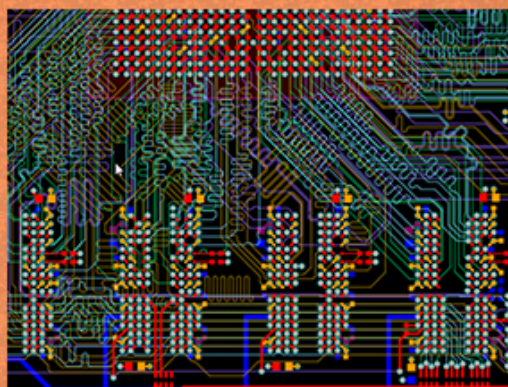
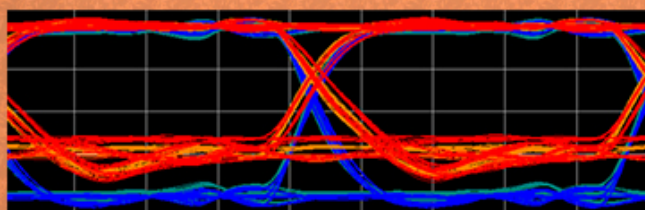




صفر تا صد حافظه پرسرعت DDR (اصول، طراحی و شبیه سازی) _ قسمت اول

صفر تا صد حافظه های پرسرعت DDR3 و DDR4 (اصول و مبانی، طراحی و شبیه سازی)



<https://blog.microele.com>



 [hosein-movahedian-attar](#)

تاریخ انتشار ۲۰ دی، ۱۴۰۰ توسط حسین موحدیان عطار

سلام به همه همراهان **مایکروالکام**. این اولین مطلب بنده در وبلاگ مایکروالکام است و قراره در قسمت های بعدی هم ادامه داشته باشد. در این سری از مطالب قرار هستش که به موضوع حافظه های موقت و پرسرعت DDR بپردازم و در هر بخش آن را تکمیل کنم. پس با من تا انتهای مطلب همراه باشید. همچنین شما میتویند سایر مطالب من رو از **این لینک** و یا آدرس Linkedin من مطالعه و مورد نقد و بررسی قرار بدید. خوشحال می شم هم از طریق Comment ذیل این



پست و هم از طریق Email و [Linkedin](#) با من در ارتباط باشید.

مقدمه

امروزه در تمام سیستم های مبتنی بر Embedded و پردازنده های پر قدرت در طراحی های مهندسان الکترونیک، استفاده از حافظه های موقت و پر سرعت DDR با نسل های مختلف بسیار متداول است. من قصد دارم با ارائه یک سلسله مطالب ضروری در بیان اصول و ساختارشناسی حافظه های DDR، طراحی و شبیه سازی، کلیه مطالب ضروری در این زمینه را برای علاقه مندان و طراحان حوزه الکترونیک شرح دهم. در مجموعه مطالب ارائه شده در بخش اول، به بیان کلیات ساختاری حافظه های DDR و تفاوت های آن ها صحبت می کنم. بعلاوه مختصی هم درباره روش های شبیه سازی و منطق حاکم بر آن مطالبی رو بیان خواهم کرد. سعی می کنم در چهار یا نهایتا پنج قسمت دیگه (مثل همین مطلب)، کلیه مطالب لازم در باب اصول و ساختار حافظه های DDR، طراحی و شبیه سازی اون ها رو بطور کامل شرح بدهم.

بخش اول: مقدمات ساختاری و مفاهیم اولیه شبیه سازی حافظه های DDR

تفاوت ساختاری حافظه های DDR3 و DDR4

استفاده از حافظه های DDR در طراحی های صنعتی بسیار مرسوم است ولی اگر امکان استفاده از هر دو دسته DDR3 و DDR4 را داشته باشیم، استفاده از کدام یک بهینه است؟

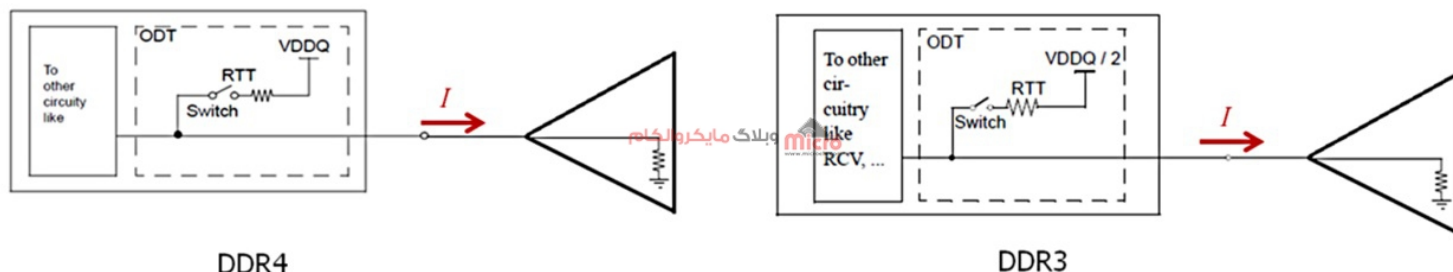
تفاوت اول، مصرف توان

حتما شنیده اید که DDR4 توان مصرفی پایین تری دارد. ولی چرا؟ ساختار پین های داده و کلاک در Byte Group های DDR4 به ساختار "Pseudo open Drain" (یا Open Drain کاذب) مشهور است.

در این ساختار، پایه داده حافظه، در حالتی که دارد یک pin با مقاومت Pull-Down را درایو می کند، جریانی برابر $I = \frac{VDDQ}{R_{tt} + R_1}$ را از خود خارج می کند. برای حافظه های DDR3، این مقدار برابر $I = \frac{VDDQ}{2} / (R_{tt} + R_1)$ است. در صورتی که VDDQ در هر دو نوع برابر باشد، جریان DDR4 بیشتر (و حدود دو برابر) DDR3 است. البته باید گفت VDDQ



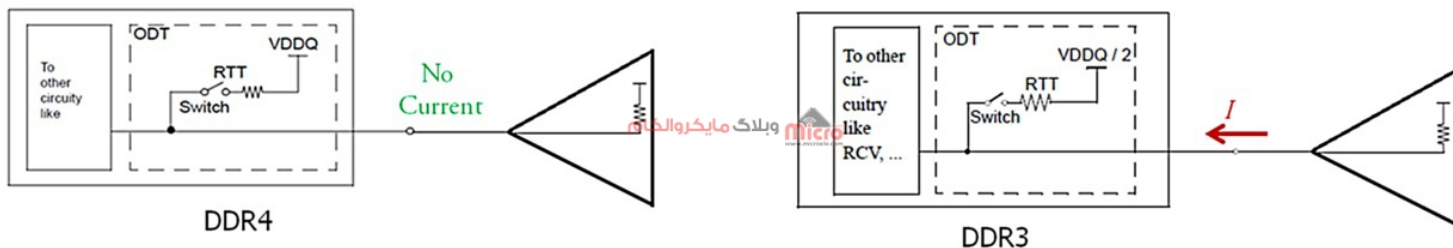
برای DDR4 کمتر از VDDQ برای DDR3 است. لذا این مقدار مطابق شکل بسیار کمتر از دو برابر است.



مدار Termination حافظه های DDR3

درایور Pin های حافظه DDR4

حال اگر Pin های حافظه DDR4 توسط یک درایور Pull-Up درایو شود، جریان ورودی به پین حافظه حدود صفر است. دلیل این امر این است که اختلاف ولتاژی بین VDDQ و ولتاژ Termination وجود ندارد. ولی برای DDR3 این جریان برابر با $I = (VDDQ/2) / (R_{tt} + R_1)$ است که در شکل زیر می توان آن را مشاهده کرد. مشخص است که مصرف توان در DDR4 با توجه به این ویژگی کمتر از DDR3 خواهد بود.



مدار Termination حافظه های DDR4

عمق این بهینگی زمانی مشخص می شود که قابلیت (Data Bus Inversion (DBI وجود دارد، بکار بسته شود. این قابلیت این امکان را فراهم می کند که اگر 5 بیت و یا بیشتر از یک داده با عرض 8 بیت، با مقدار صفر درایو شده بودند، همه بیت ها به شکل Invert شود و در عوض، یک پایه به نام DBI یک گردد تا به حافظه بفهماند که "من درایور، چنین بلایی بر سر داده آورده ام! حواست باشد!!!"

در این حالت، هیچگاه بیش از 4 پایه از 9 پایه (8 عدد DQ و یکی DBI)، مقدار صفر ندارد. لذا کمترین تعداد پایه، همواره در حالت رای صفر قرار می گیرند. لذا کمترین پایه همواره در حالت مصرف توان قرار می گیرند! و این یعنی توان مصرفی پایین تر در DDR4.

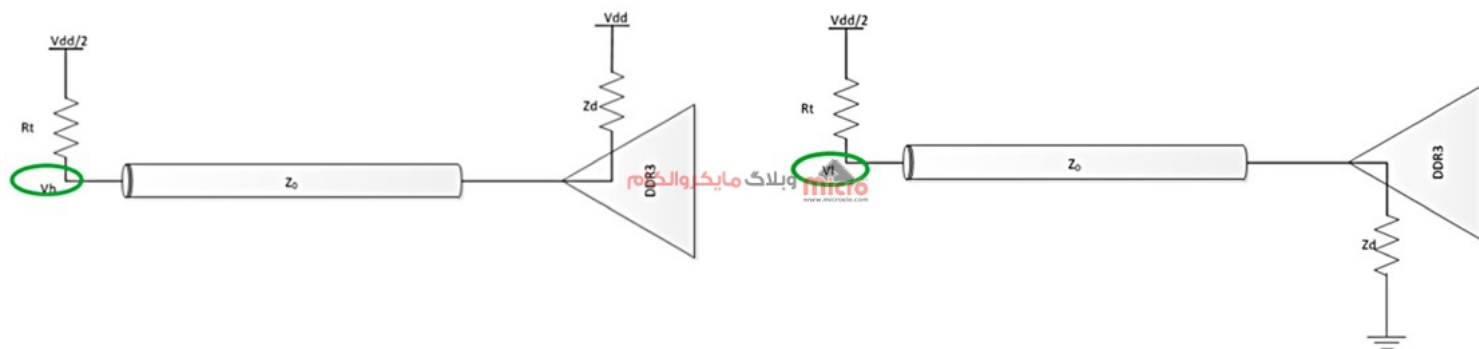


قابلیت (Data Bus Inversion) (DBI) در حافظه های DDR4

تفاوت دوم، ولتاژ رفرنس:

در حافظه های DDR3، برای این که مشخص کند که یک Pin داده در وضعیت صفر قرار دارد و یا در وضعیت یک، از بیرون تراشه، یک ولتاژ مرجع به حافظه داده می شود. در حالی که در DDR4، این ولتاژ در داخل خود حافظه DDR4 تولید شده و قابل تنظیم است و باید بعد از هر Power up مجدداً تنظیم گردد.

برای تعیین مقدار این ولتاژ مرجع، باید میانگین ولتاژ در مقصد داده یعنی گیرنده را در حالتی که Drive و Drive Low High می شود، اندازه گیری کنیم. در DDR3 به شکل زیر عمل می کنیم.



ولتاژ مرجع در حافظه های DDR3

همان طور که دیده می شود، می توان میانگین را به صورت زیر محاسبه کرد. هطبق روابط زیر، ولتاژ میانگین، هیچ ارتباطی به مبدا، مقصد و شرایط محیطی آنها ندارد و مستقل از همه موارد مذکور، همواره ثابت و برابر نصف Vdd است.



$$V_h = (V_{dd}) \left(\frac{R_t}{R_t + Z_d} \right) + \left(\frac{V_{dd}}{2} \right) \left(\frac{Z_d}{R_t + Z_d} \right)$$

$$V_h = \left(\frac{V_{dd}}{2} \right) \left(\frac{2R_t + Z_d}{R_t + Z_d} \right)$$

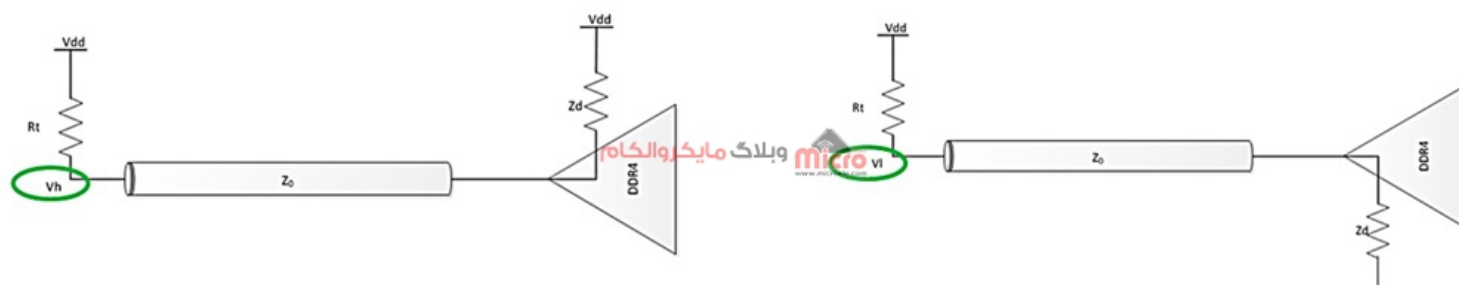
$$V_l = \left(\frac{V_{dd}}{2} \right) \left(\frac{Z_d}{R_t + Z_d} \right)$$

$$V_{avg} = \frac{V_h + V_l}{2} = \left[\left(\frac{V_{dd}}{2} \right) \left(\frac{2R_t + Z_d}{R_t + Z_d} \right) + \left(\frac{V_{dd}}{2} \right) \left(\frac{Z_d}{R_t + Z_d} \right) \right] / 2 = \left[\left(\frac{V_{dd}}{2} \right) \left(\frac{2R_t + 2Z_d}{R_t + Z_d} \right) \right] / 2$$

$$= \frac{V_{dd}}{2}$$

میانین گیری ولتاژ در DDR3

ولی در DDR4 داریم:



ولتاژ مرجع در DDR4

که در آن ولتاژ میانگین به صورت زیر بدست می آید.

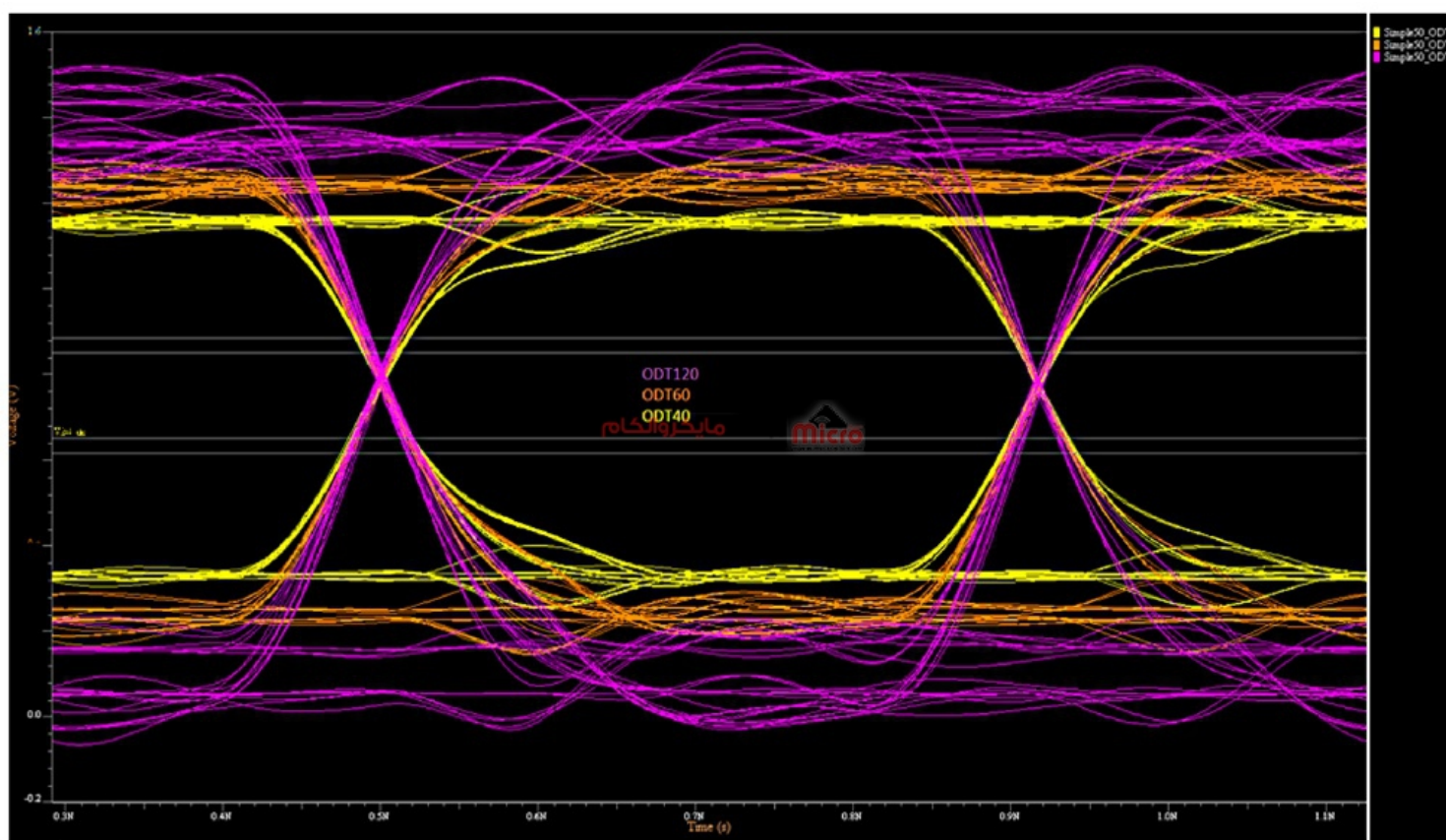
$$V_{avg} = \frac{V_h + V_l}{2} = \left[(V_{dd}) + (V_{dd}) \left(\frac{Z_d}{R_t + Z_d} \right) \right] / 2$$

$$= \left(\frac{V_{dd}}{2} \right) + \left(\frac{V_{dd}}{2} \right) \left(\frac{Z_d}{R_t + Z_d} \right)$$



میانگین ولتاژ در DDR4

همان طور که دیده می‌شود، مقدار آن به مقاومت Termination در مصد داده و مقاومت Pull-Down در مبدا داده وابسته است. حال اگر در حافظه DDR4، مقدار مقاومت Termination در گیرنده را زیاد کنیم، فقط سطح بالایی و پایینی سیگنال از دو طرف افزایش می‌یابد ولی در همه حال، وسط سیگنال، مقداری ثابت دارد. در زیر این کار را با افزودن مقاومت Termination از 40 تا 80 اهم را برای DDR3 مشاهده می‌کنید.



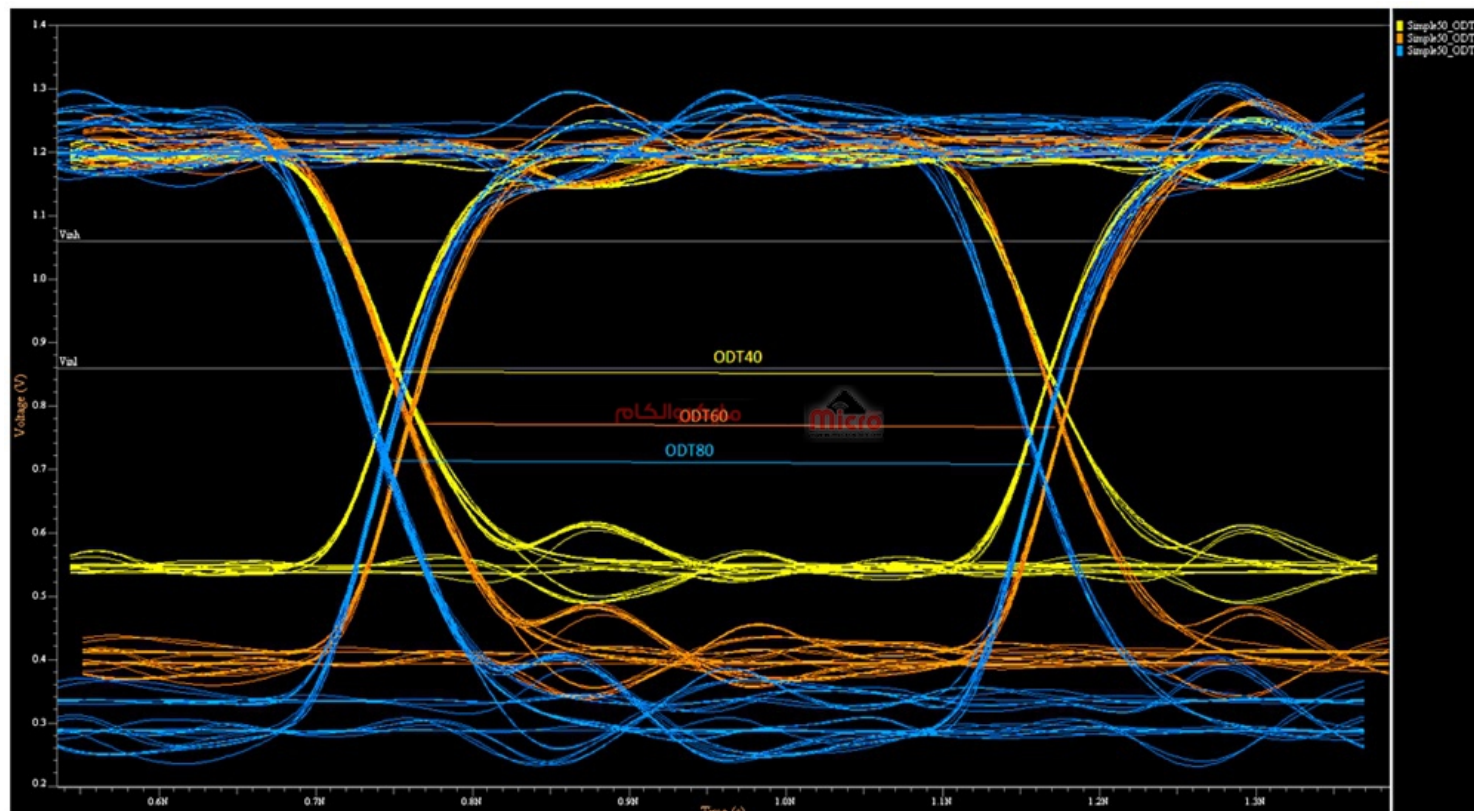
تاثیر مقاومت Termination در حافظه DDR3

وابستگی سطح ولتاژ مرجع به مقدار Termination

در DDR4، به دلیل وابستگی سطح ولتاژ مرجع به مقدار Termination در گیرنده، در صورت افزودن مقدار مقاومت Termination مقصد داده، فقط این سطوح سیگنال نیستند که از هم فاصله می‌گیرند، بلکه (با توجه فرمول)، ولتاژ مرجع سیگنال نیز کاهش می‌یابد. این را در شکل زیر با افزودن مقاومت Termination در مقصد داده، از 40 به 80 اهم



می بینید.



تأثیر مقاومت Termination در حافظه DDR4

مشخصاً دو سر یک Lane داده، باید Terminate شده باشد تا داده در محل Receiver دریافت گردد و همچنین Antenna نیز نشود. در این جا بحث بسیار مهمی مطرح می شود و آن این که اگر ولتاژ مرجع، خودش در حال جابجا شدن باشد، این اولاً خوب است یا بد؟ ثانیاً آیا ولتاژ مرجعی که خودش در حال تغییر است، قابل اطمینان است یا خیر؟ (اصطلاحاً آیا دیگه سنگ روی سنگ بند می شه؟) در جواب باید ببینیم ولتاژ مرجع، در DDR4 چگونه و طی چه مراحل اندازه گیری می شود؟

روش های اندازه گیری ولتاژ مرجع در حافظه DDR4

برای حافظه DDR4، ولتاژ مرجع را می توان با دو روش تخمین زد. Option1 و Option2 (از خودم در نیاوردم! واقعا همین نام های عجیب رو داره!!!)



- در Option1، روش کار به این صورت است که متوسط ولتاژ مرجع تمام سیگنال ها را به عنوان ولتاژ مرجع تمام Pin ها در نظر می گیرند.

$$V_d = \frac{\sum V_x}{n}. \text{ (Device Vref is average of all pins)}$$

اندازه گیری ولتاژ مرجع در حافظه DDR4 در روش Option1

- در Option2، روش کار به این صورت است که فقط متوسط ولتاژ های مرجع extreme (کمینه و بیشینه) را بدست می آورند و از همان به عنوان ولتاژ مرجع کل سیگنال ها استفاده می کنند. برای مثال اگر ولتاژ مرجع برای هشت Pin داده به ترتیب زیر باشد، داریم:

800mV, 750mV, 730mV, 725mV, 720mV, 710mV, 705mV and 700mV

$$\text{Option 1: } \frac{800+750+730+725+720+710+705+700}{8} = 730mV$$

$$\text{Option 2: } \frac{800+700}{2} = 750mV$$

اندازه گیری ولتاژ مرجع در حافظه DDR4 در روش Option2

سوال: کدام بهتر است؟

حافظه های DDR4، در روال شبیه سازی و یا Config شان، این امکان را فراهم می کنند که خودتون بین این دو گزینه انتخاب کنید. ولی من شخصا Option2 را ترجیح می دهم. (چند لحظه دیگر، شما هم به همین نتیجه می رسید!). اگر V_x را ولتاژ مرجع سیگنال x در نظر بگیرید و V_d را ولتاژ مرجعی که می خواهید برای تمام سیگنال ها از آن استفاده کنید در نظر بگیرید، این دو ولتاژ باید کمترین اختلاف را با هم داشته باشند. یا اختلافشان معقول باشد. به این معنی که اگر ولتاژ مرجع سیگنال x را نیز همان V_d در نظر گرفتید، سیگنال x به مشکل نخورد و تغییرات High و Low آن، به اندازه ای باشد که ولتاژ مرجع را در هر بار Toggle بگذرانند و DDR4 بفهمد که سیگنال High شده یا Low شده است.



سطوح ولتاژ در DDR4

سوال: این مقدار معقول تفاوت بین V_d و V_x چقدر است؟

اگر به شکل بالا نگاه کنید، متوجه می شوید که عملاً با جایگزینی V_x با V_d در فرایند تشخیص سطوح Valid سیگنال های داده، قدرت تشخیص ما برای High یا Low شدن سیگنال کم می شود. یعنی اگر قبلاً (زمانی که V_x مرجع سیگنال x بود)، با عبور سیگنال x از $V_{x-min-high}$ ، تشخیص می دادیم که سیگنال x، به حالت High درآمده است، حالا کمی کارمان سخت تر شده و باید سیگنال x از $V_{d-min-high}$ عبور کند تا DDR4 بفهمد که سیگنال x به حالت High درآمده است. (عجب گرفتاری شدیم!)

به این حادثه ناگوار، Margin Loss (یا از دست دادن بخشی از حاشیه اطمینان سیگنال) می گوئیم. مقدار Margin Loss به صورت زیر بدست می آید:

$$Margin\ Loss = M_p - M_d = V_d - V_x$$

مقدار Margin Loss

- البته گاهی هم ممکن است Margin Loss مقدار منفی داشته باشد که به آن Margin Gain می گوئیم و نه تنها بد نیست، بلکه خوب هم هست. و آنجایی اتفاق می افتد که V_x سیگنال x از V_d کل سیگنال ها بیشتر باشد. که البته موضوع بحث ما نیست.

محاسبه V_d

- در Option1، ولتاژ مرجع کل DDR4 میانگین ولتاژ های مرجع کل سیگنال ها است.



$$V_d = \frac{\sum V_x}{n}. \text{ (Device Vref is average of all pins)}$$

اندازه گیری ولتاژ مرجع در حافظه DDR4

- در همین Option1، مقدار Margin Loss متوسط کل سیگنال ها از فرمول زیر محاسبه می شود. البته کاملاً مشخص هم بود! متوسط Margin Loss متوسط در Option1 همواره صفر است و Margin Loss یک سیگنال توسط Margin Gain بقیه سیگنال های DDR4 جبران می شود.

$$\text{Average Margin Loss} = \frac{\sum (V_d - V_x)}{n} = \frac{\sum V_d}{n} - \frac{\sum V_x}{n} = \frac{nV_d}{n} - V_d = 0$$

مقدار Margin Loss در Option1

سوال: این خیلی خوبه که، چرا پس گزینه مطلوب من، باید Option2 باشه؟

جواب: صبر کنید... الان می بینید چرا!

- برای Option2 داریم:

$$V_d = \frac{V_{max} + V_{min}}{2}. \text{ (Device Vref is average of extreme pins)}$$

اندازه گیری ولتاژ مرجع در حافظه

- Margin Loss متوسط کل DDR4:



$$\begin{aligned} \text{Average Margin Loss} &= \frac{\sum (V_d - V_x)}{n} = \frac{(V_d - V_{\max}) + (V_d - V_{\min})}{n} + \frac{\sum_{x \neq \max, \min} (V_d - V_x)}{n} \\ &= \frac{2}{n} \left(\frac{V_{\max} + V_{\min}}{2} \right) - \frac{(V_{\max} + V_{\min})}{n} + \frac{\sum_{x \neq \max, \min} (V_d - V_x)}{n} \\ &= \frac{\sum_{x \neq \max, \min} (V_d - V_x)}{n} \end{aligned}$$

Margin Loss متوسط کل DDR4

بله، همان طور که می بینید در Option2، مقدار Margin Loss صفر نیست یا اصطلاحاً کمی از حاشیه ایمن تشخیص سیگنال را از دست داده ایم. اگر چه به نظر می رسد که Option1 بهتر باشد، ولی اینطور نیست و Option2 بهتر است. دلیل آن هم با در نظر گرفتن سیگنال های Maximum و Extreme (Minimum) در مجموعه سیگنال های DDR4 نهفته است.

مثال

حالا با یک مثال منظور خودم را می گویم. فرض کنید یک سیگنال x ای داریم (مثل همونی که اون اول مثال زدم و بین 930mV و 670mV در حال Swing است. پس V_x آن برابر با 800mV است). Option1 مقدار V_d اون رو 730mV تخمین زده و Option2 مقدار V_d اون رو 750mV تخمین زده است. حالا اگر اجازه بدید، مقدار V_p-p بین $V_d-\max-\text{Low}$ و $V_d-\min-\text{High}$ رو حدود 260mV در نظر بگیریم.

خوب با این اوصاف Option1 مقادیر $V_d-\max-\text{Low}$ و $V_d-\min-\text{High}$ رو به ترتیب 662mV و 798mV محاسبه می کند. در حالی که Option2 مقادیر $V_d-\max-\text{Low}$ و $V_d-\min-\text{High}$ رو به ترتیب 682mV و 818mV محاسبه می کند. خوب حالا می بینید که سیگنال Extreme (که همون سیگنال x ما بود و از بین سیگنال ها پایین ترین سطوح ولتاژی رو داشت) Option1 رو Pass نمی کنه چون min اون، بیشتر از Min مرجع است. یا به عبارتی DDR4 اصلاً متوجه Low شدن آن نمی شود. این درحالی است که Option2، با وجود داشتن Margin کمی که برای سیگنال های Valid داره، ولی حداقل تمام سیگنال ها اون رو Pass می کنند. (چون Maximum و Minimum های سیگنال ها اون رو Pass می کنند. پس بقیه هم حتماً Pass می کنند).



Option 1

$$730 + 68 = 798\text{mV (Vinh)}$$

$$730 - 68 = 662\text{mV (Vinl)}$$

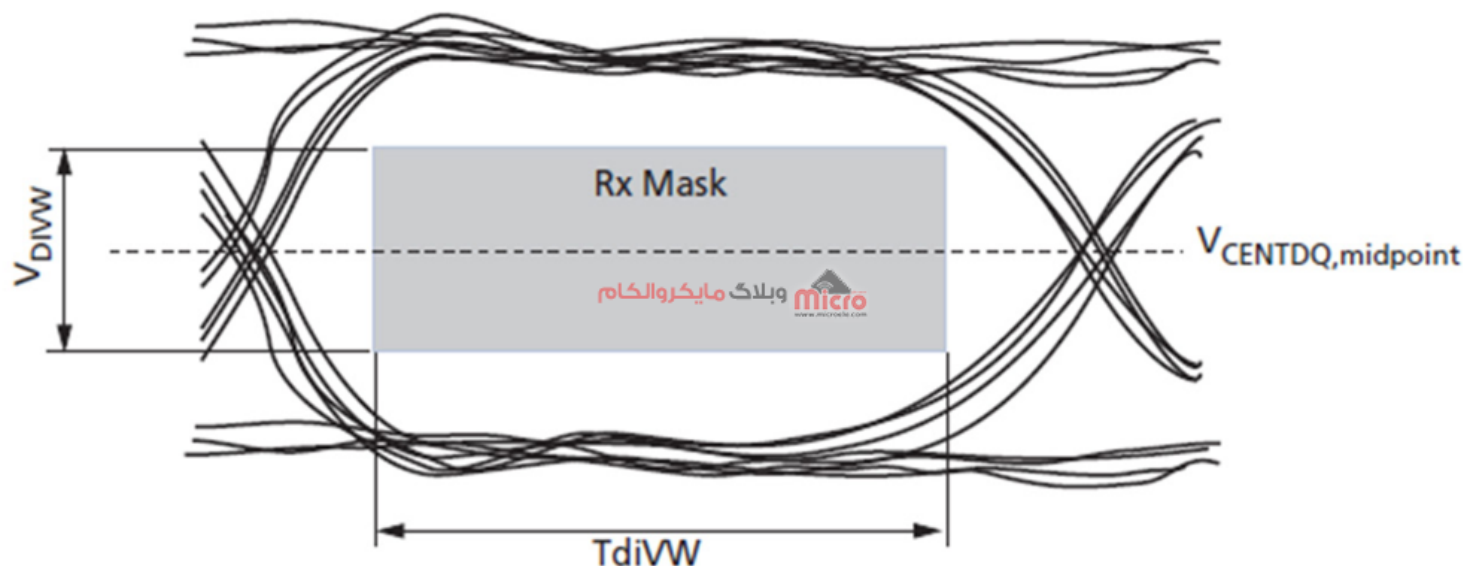
Option 2

$$750 + 68 = 818\text{mV (Vinh)}$$

$$750 - 68 = 682\text{mV (Vinl)}$$

Option 1 و Option 2 در DDR

حالا اگر کمترین مقدار بین $V_{x-\max-\text{Low}}$ تمام Pin ها را با نام V_l و بیشترین مقدار بین $V_{x-\min-\text{High}}$ تمام Pin ها را با نام V_h نام گذاری کنیم، ارتفاع قابل قبول دیاگرام چشمی (Eye Diagram)، برابر با $V_{DIVW} = V_h - V_l$ است.



ارتفاع قابل قبول دیاگرام چشمی (Eye Diagram)

پیشنهاد

یه پیشنهاد دوستانه. اگر سیگنال شما در سمت Receiver دارای Eye Diagram قابل قبول است، گیر ندید به سیگنال سمت



Transmitter! اون گاهی قاطی داره. اصلا هم مهم نیست. برآیند کار شما در شبیه سازی، همون سیگنال سمت Receiver است و بس.

روش های شبیه سازی سیگنال های حافظه DDR4

حافظه DDR4 دارای کلاک مسیل با داده نیست. کلاک داده با پین جداگانه ای بنام DQS به همراه Byte Group منتقل می شود. (لطفا با کلاک Address ها اون رو قاطی نکنید). ضمنا حافظه DDR4 شبیه سازی رو از SerDes ها وام گرفته. اون اوایل اصلا شبیه سازی واسه حافظه DDR4 مطرح نبود.

شبیه سازی Eye Diagram مربوط به پین های داده (DQ) تراشه DDR4. نباید فقط با استفاده از یک Period ثابت سیگنال DQ انجام بپذیرد. و حتما باید بنا به کلاک مربوط به خودش (DQS)، انجام شود. لذا این دو (DQ و DQS) هر کدام ناهنجاری داشته باشند، خود را مشخصا در Eye Diagram نشان می دهد. کمی موضوع را باز می کنم. فرض کنید سیگنال DQ را داریم نمونه برداری می کنیم. خود سیگنال هیچ jitter ای ندارد. (یعنی از نظر زمانی جابجا نمی شود و دوره پریود آن ثابت می ماند).

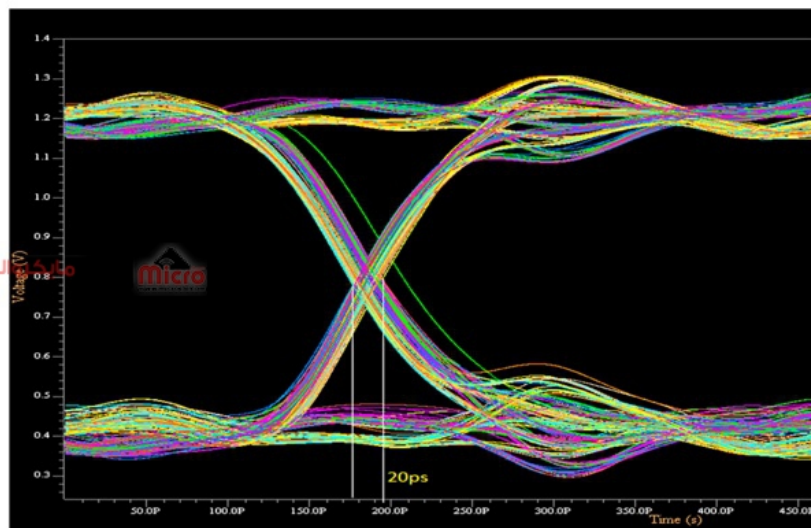
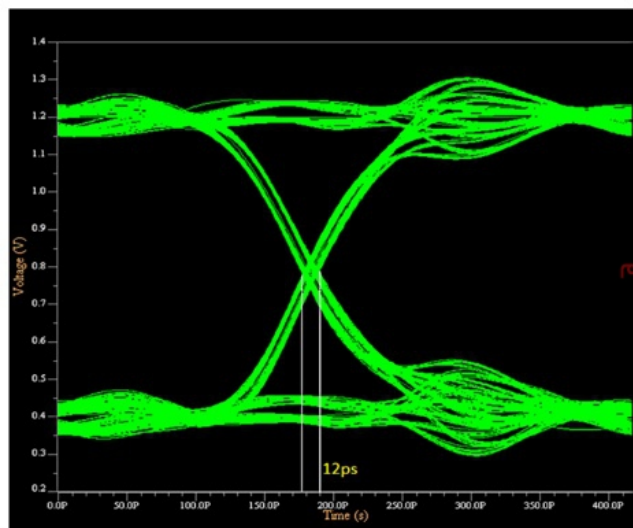
رخ دادن Jitter در کلاک

حال اگر کلاک دچار Jitter شود، یک بار سیگنال را در وسط آن نمونه برداری می کند و یک بار کمی عقب تر و یک بار کمی جلو تر و به همین ترتیب. حالا این نمونه ها را روی هم می اندازد تا Eye Diagram را بسازد. نتیجه فاجعه است! Eye Diagram به نظر می رسد دارد راه می رود و Jitter در آن مشهود است. برعکس موضوع نیز صادق است. اگر DQS دچار Jitter نباشد ولی خود سیگنال DQ دچار Jitter باشد، همین آش است و همین کاسه! حالا اگر هر دو دچار Jitter باشند، دیگه واویلا. لذا نمونه برداری گاهی زمانی انجام می شود که از آن Period قبلا نمونه برداری انجام شده و گاهی هم اصلا نمونه ای از یک برحه از زمان برداشته نمی شود چون بنا به Period زمانی نمونه برداری آنچنان نمی شود. خوب، انگار خیلی بد شد!

گاهی از یک بخش از شکل موج، دوتا نمونه داریم. گاهی از بخشی دیگر هیچ نمونه ای نداریم. انگار خیلی آشوب و بلوا پیاپی شد. نه، اینطور نیست. راهکار هایی وجود دارد تا اوضاع بهتر شود. اول این که اگر سیگنالی را زود نمونه برداری کرده ایم می توانیم آن را شیفت زمانی بدهیم. ولی راهکاری برای آن داده های از دست رفته نداریم. DDR4 های در عمل این کار را انجام می دهند و این شیفت DQS، نمونه برداری سیگنال را تحت تاثیر قرار می دهد.



مثلا در زیر، می‌توان نمونه برداری از سیگنال DQ با سیگنال DQS بدون Jitter (سمت چپ) و نمونه برداری از سیگنال DQ با سیگنال DQS دچار Jitter را ببینید. در سمت راست (DQS دچار Jitter)، اصلا انگار سیگنال در برخی مواقع، دیر نمونه برداری شده است.



نمونه برداری از سیگنال DQ با سیگنال DQS بدون Jitter (سمت چپ) و نمونه برداری از سیگنال DQ با سیگنال DQS دچار Jitter

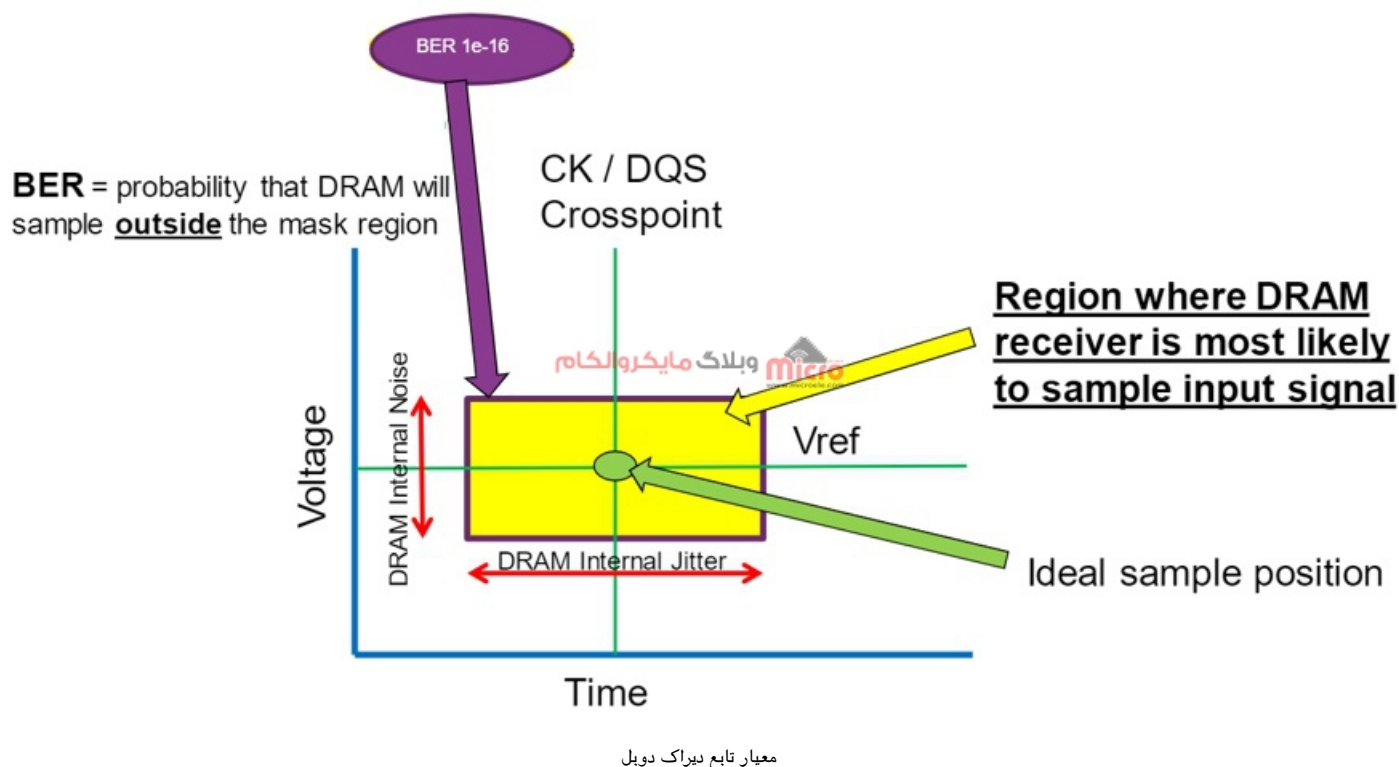


مخلص کلام

خلاصه، اگر بخواهیم Eye Diagram برای DDR4 به اندازه قابل قبولی باز باشد، باید Jitter و Noise آن در یک محدوده قابل قبولی باشد. و یا به عبارتی از یک محدوده ای کمتر نشود یا به زبان دیگر، دیاگرام چشمی، به اندازه خاصی باز باشد. چون Reference Voltage برای هر تراشه DDR4 ای ممکن است متفاوت باشد. لذا نمی توان الگوی عددی ثابت با مکان ثابت در Eye Diagram ارائه داد. ولی می توان الگوی ثابتی با مکان متغیر در Eye Diagram ارائه داد.

معیار تابع دیراک دوبل (Dual-Dirac Model) برای شبیه سازی DDR

JESD79-4 یک استاندارد برای DDR4 است و اجبار می کند که BER (Bit Error Rate) برای DDR4 حداکثر باید $1E6$ (برابر با 10^{16} خطا در ثانیه) باشد. این خطا را با معیار تابع دیراک دوبل (Dual-Dirac Model) بدست آورده که یک ابزار کارآمد برای تخمین Jitter در سینک های سریال است. (یه چیزی توی این مایه ها که می گه اگر می خوای Jitter در محدوده مجاز استاندارد باشه، باید هر 106 بیت یکبار خطا در نمونه برداری داشته باشی. خب این خیلی سخته، احتمالا برای هیچ کسی ممکن نیست اینقدر صبر کنه).



معیار تابع دیراک دوپل

الگوی بدترین حالت (Worst-Case Bit Pattern) برای شبیه سازی DDR

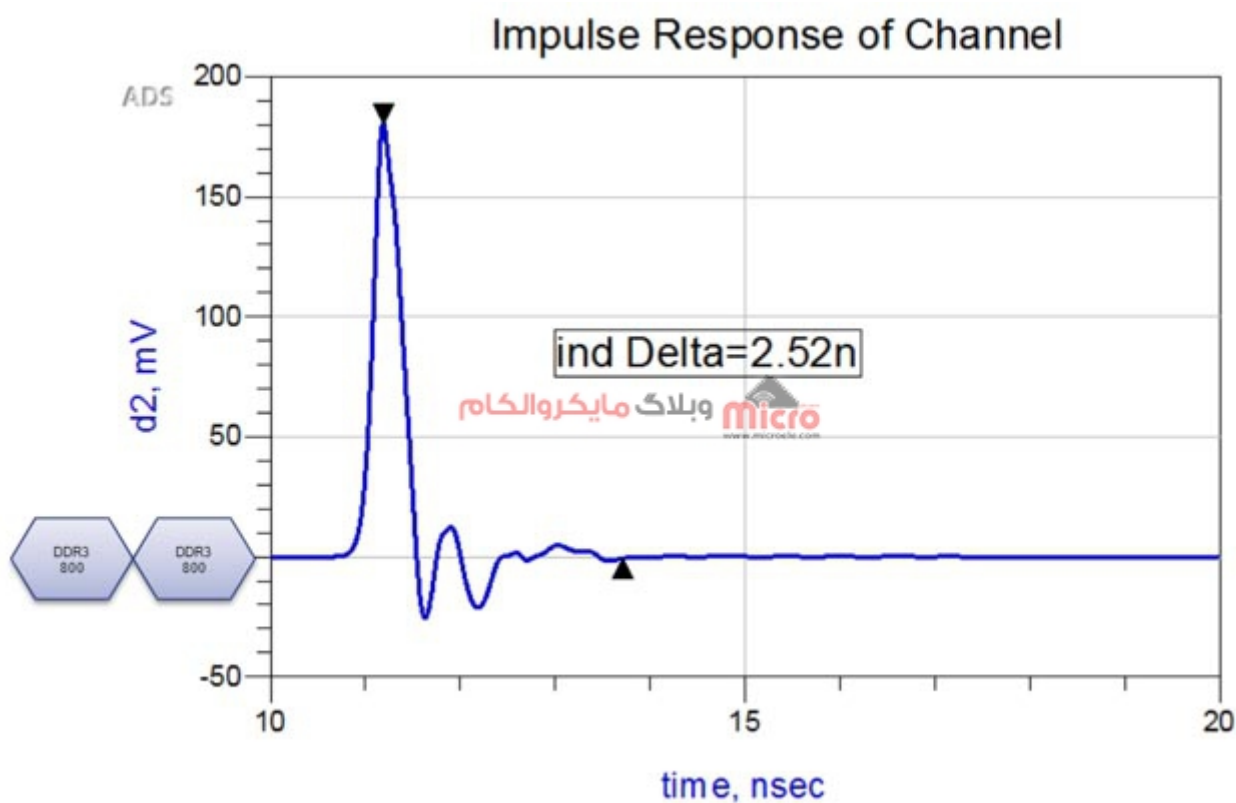
روش دوم استفاده از "الگوی بدترین حالت" (یا همان Worst-Case Bit Pattern) است. اگر پاسخ ضربه سیستم یادتون باشه، درک این روش براتون راحت تر خواهد شد. دقیقاً مثل پاسخ ضربه. ولی این دفعه پاسخ پالس کانال DDR ها رو اندازه می گیریم. مثلاً اگر DDR ما سه تا Slot داشته باشه، پاسخ کانال داده را به پالس مشخصی اندازه می گیریم و دو سه تا پارامتر رو از روی اون بدست میاریم.

مثال

یک DDR با سه Slot از DDR، روی کانال داده (شماتیک نیست. شماتیک شبیه سازی هست، فعلاً از من قبول کنید. در آینده توضیح خواهم داد چطور بسازیدش!)



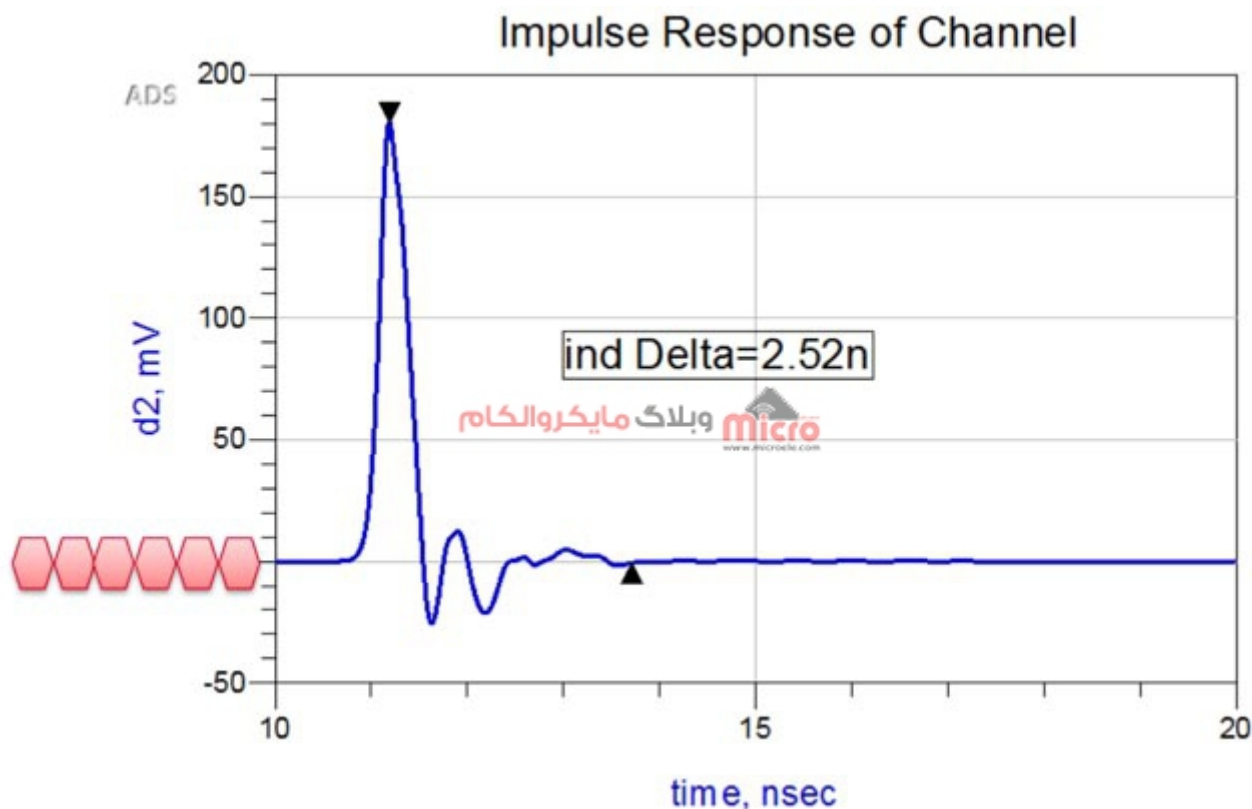
- مثلاً برای DDR3 800MHz، دو تا بیت داده با نرخ خود DDR به داخل کانال داده می فرستیم. زمان ورود بیت سوم، بیت اول کانال رو ترک کرده. لذا با جایگشت 2^8 احتمال وقوع بدترین حالت وجود دارد.



پاسخ ضربه $DDR3$



- مثلاً برای DDR4 2400MHz، شش تا بیت داده با نرخ خود DDR به داخل کانال داده می فرستیم. زمان ورود بیت هفتم، بیت اول کانال رو ترک کرده. لذا با جایگشت 2^6 احتمال وقوع بدترین حالت وجود دارد.



پاسخ ضربه‌ای به DDR4

داستان داره، اینقدر ها هم ساده نیست. ما اثر Crosstalk کانال و QD ها روی هم رو اصلاً در نظر نگرفتیم. حالا اگر اون رو اضافه کنیم، زمان خارج شدن اثر پالس از کانال هر کدام از DQ ها، از DQ بقلی خودش، زمان رو طولانی می کنه و تعداد بیت مورد نیاز برای رخدادن بدترین حالت رو بیشتر و در نهایت هم باعث افزایش زمان شبیه سازی خواهد شد.

بس بیایید دست به دست هم بدیم به مهر و با طراحی صحیح PCB، اثر Crosstalk رو به حداقل برسونیم.

نکته

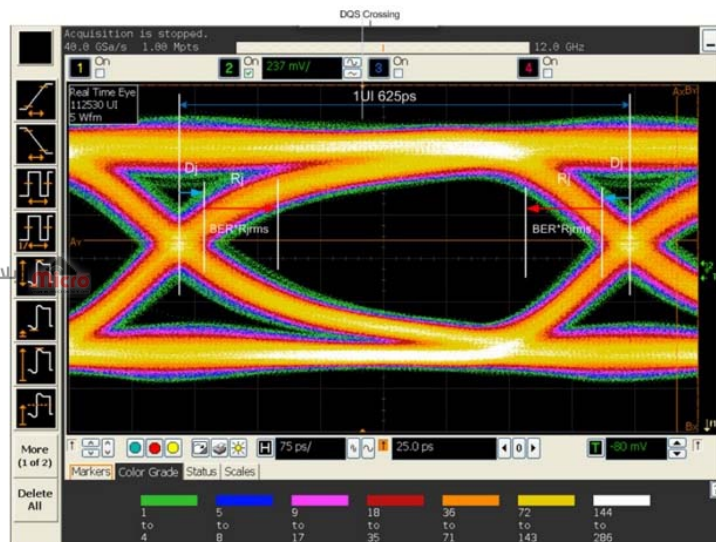
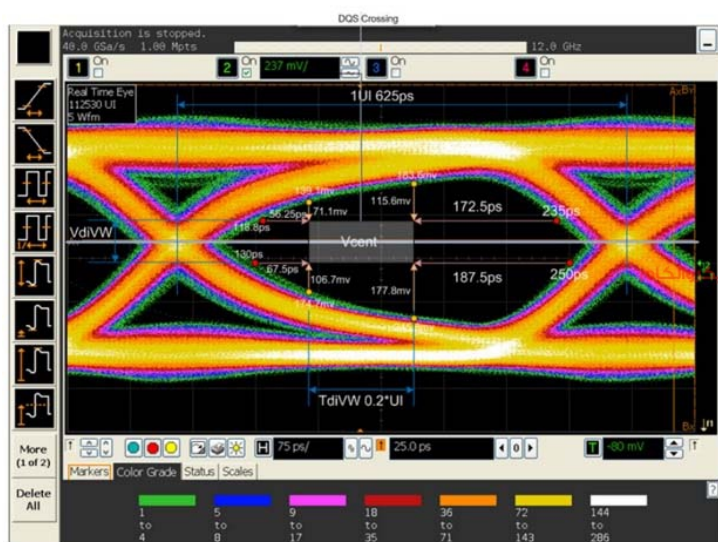
اگر مطمئن نیستید که Worst-Case Bit Pattern را گرفته اید، این روش را استفاده نکنید چون احتمالاً همچنان مدار شما شامل افزایش (RJ) Random Jitter می شود. (اگر می خواهید بدونید RJ چیه، شکل زیر رو ببینید.) Jitter دو قسمت داره.



بخش رندوم و بخش قابل پیشبینی. این توی همون مدل دلتای دوپل دیراک تعریف می‌شود. که از حوصله بحث خارج هست.

فقط بدونید که بخش (DJ Deterministic Jitter) قابل پیشبینی است و با خطای نمونه برداری هیچ افزایش و کاهشی نداره. ولی (RJ Random Jitter) قطعا تا زمانی که به بدترین حالت پاسخ پالس کانال دست نیافته باشیم، احتمال داره تغییر کند. واسه این، تعریف این دو تا رو الآن مطرح کردم که تا ندونید به چه دردی می‌خوره، چه فایده که بگم چی هستند.

در شکل سمت چپ، این اندازه گیری بیش از 10000 بار UI، منجر به TdVW اندازه گیری شده برابر با 353.57ps یا به تعبیر دیگه، حاشیه 56.25ps برای ماسک شده و VdVW اندازه گیری شده برابر با 377mv یا به عبارتی حاشیه 71.1mv می‌شود. در شکل سمت راست، تعداد تکرار را به 10^{12} بار (یعنی یک تریلیون بار تکرار) UI افزایش دادیم و نهایتا مشخص شد که با این تعداد تکرار، بالاخره شکل موج سیگنال، با مربع معیار، تداخل پیدا کرده است. پس می‌بینید که این روش هم ممکن است جواب نده.



شکل سمت چپ، این اندازه گیری بیش از 10000 بار UI. شکل سمت راست، تعداد تکرار را به 10^{12} بار (یعنی یک تریلیون بار تکرار) UI

سوال

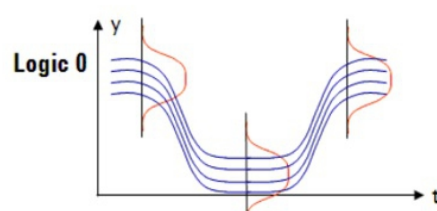
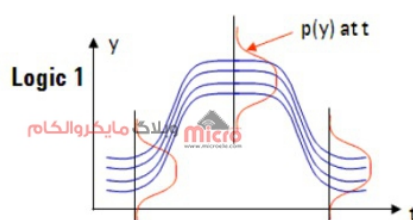
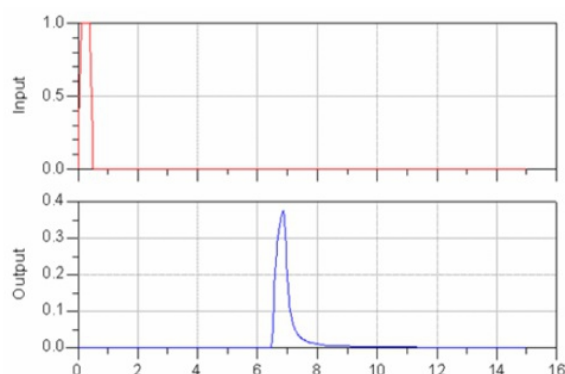
حالا چیکار کنیم؟ چطوری حداقل محیط لازم برای دیگرام چشمی رو در چند ثانیه (نه چند روز و چند ماه و چند سال) بدست بیاریم؟



پاسخ ورودی و خروجی Driver و DDR ها به ورودی بیت های رندوم، در شبیه سازی DDR

در پاسخ سوال قبل، روش پیشنهادی استفاده از بیت های Random و گرفتن پاسخ در ورودی و خروجی های DDR و Driver است. برای این روش، یک شبیه سازی گذرا کوتاه را اجرا کنید تا پاسخ ضربه ای کانال، Rx, Tx را دریافت کنید و بعد معیار Eye Diagram را از پاسخ ضربه و ویژگی های تصادفی یک الگوی بیت غیر تکرار بی نهایت بسازید. (جمله ام آنقدر هم که فکر می کنید، بی پایه و ساده نیست، الان توضیح می دم!)

دوستان، چرا تا الان به الگوی پالس Random فکر نکرده بودید؟ داستان همین جاست. Random بودن ورودی، احتمال رخداد RJ رو افزایش می دهد. لذا احتمال این که در زمان به مراتب کمتر، به "رخداد تصادفی خطای نمونه برداری" برسیم، فوق العاده بیشتر از قبل می شود. (دوستانی که آمار رو 20 گرفتند، توی هوا حرف منو گرفتند!). شک نکنید هیچ راه دیگه ای وجود نداره. کلی دانشمند نشستن فکر کردن به این نتیجه رسیدن. لذا کلیه نرم افزار های شبیه سازی هم برای شبیه سازی سیگنال های سریال پرسرعت، از ورودی رندوم استفاده می کنند.

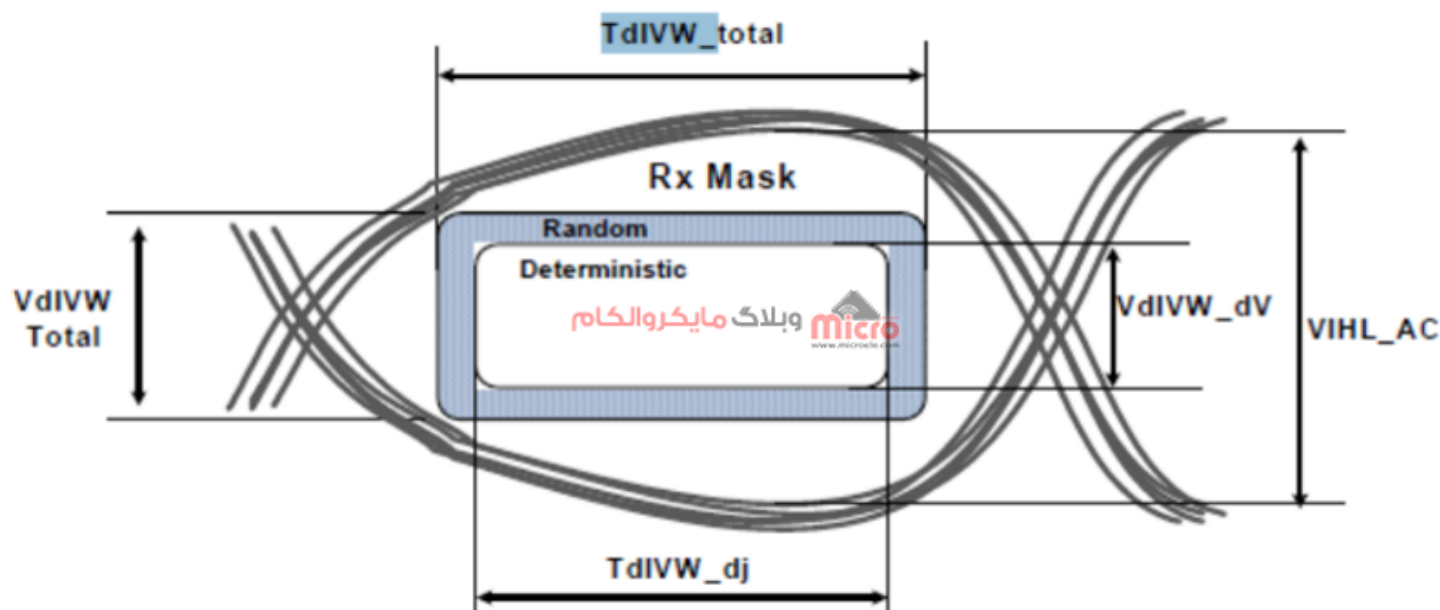


پاسخ ورودی و خروجی Driver و DDR ها به ورودی بیت های رندوم، در شبیه سازی DDR

فقط یه نکته ای رو بگم بدانید و آگاه باشید که اگر تصور می کنید، با روی هم انداختن دو تا نمودار توزیع Jitter بالا، نهایتاً پاسخ شما در ورودی و خروجی یکسان خواهد شد سخت در اشتباه هستید. دو تا نمودار بالا، پاسخ ورودی و خروجی به ازای پالس در ورودی است. پاسخ ورودی و خروجی به ازای پالس در خروجی، چیز دیگری است. اگر این جمله رو قبول ندارید اشکالی نداره، تا همین جا هم خیلی خوب پیش رفتیم.

سوال

TdiVW و VdiVW را برای BER های سختگیرانه تر از 10^{-16} BER، چطوری اندازه گیری کنیم؟



TdiVW

در واقع این دغدغه اون زمانی به وجود میاد که بخواهیم اون محدوده مستطیلی شکل رو برای افزایش قابلیت اطمینان تراشه (کاهش احتمال Fail شدن بیت های داده) با مقادیر سختگیرانه تر از استاندارد (یعنی $BER=10^{-16}$) بدست بیاریم. برای این کار باید بدونیم که این VdiVW و TdiVW برابر با اختلاف بین دو مقدار VdiVW_dV و VdiVW_total و اختلاف بین دو مقدار TdiVW_dj و TdiVW_total بدست می آید. البته مقادیر Jitter و Noise ثابت TdiVW_dj و VdiVW_dV، در جدول Ac Parameters تراشه DDR موجود هستند.

Symbol	Parameter	1600,1866,2133		2400		2666,3200		Unit	NOTE
		min	max	min	max	min	max		
VdiVW_total	Rx Mask voltage - p-p total	-	136 (note12)	-	tbd	-	tbd	mV	1,2,4,6
VdiVW_dV	Rx Mask voltage - deterministic	-	136 (note12)	-	tbd	-	tbd	mV	1,5,13
TdiVW_total	Rx timing window total	-	0.2 (note12)	-	tbd	-	tbd	UI*	1,2,4,6
TdiVW_dj	Rx deterministic timing	-	0.2	-	tbd	-	tbd	UI*	1,5, 13
VIHL_AC	DQ AC input swing pk-pk	186	-	tbd	-	tbd	-	mV	7

مثالی از جدول پارامترهای VdiVW_dV و VdiVW_total در یک DDR نمونه

اگر مقدار حاصل این تفاضل ها را بر ضریب کیفیت (Q Factor) که برای $BER=10^{-16}$ برابر با 8.2 است) تقسیم کنیم، Random Noise (RN) و Random Jitter (RJ) بدست خواهد آمد. در نهایت شما اگر مقدار این Random Noise (RN) و



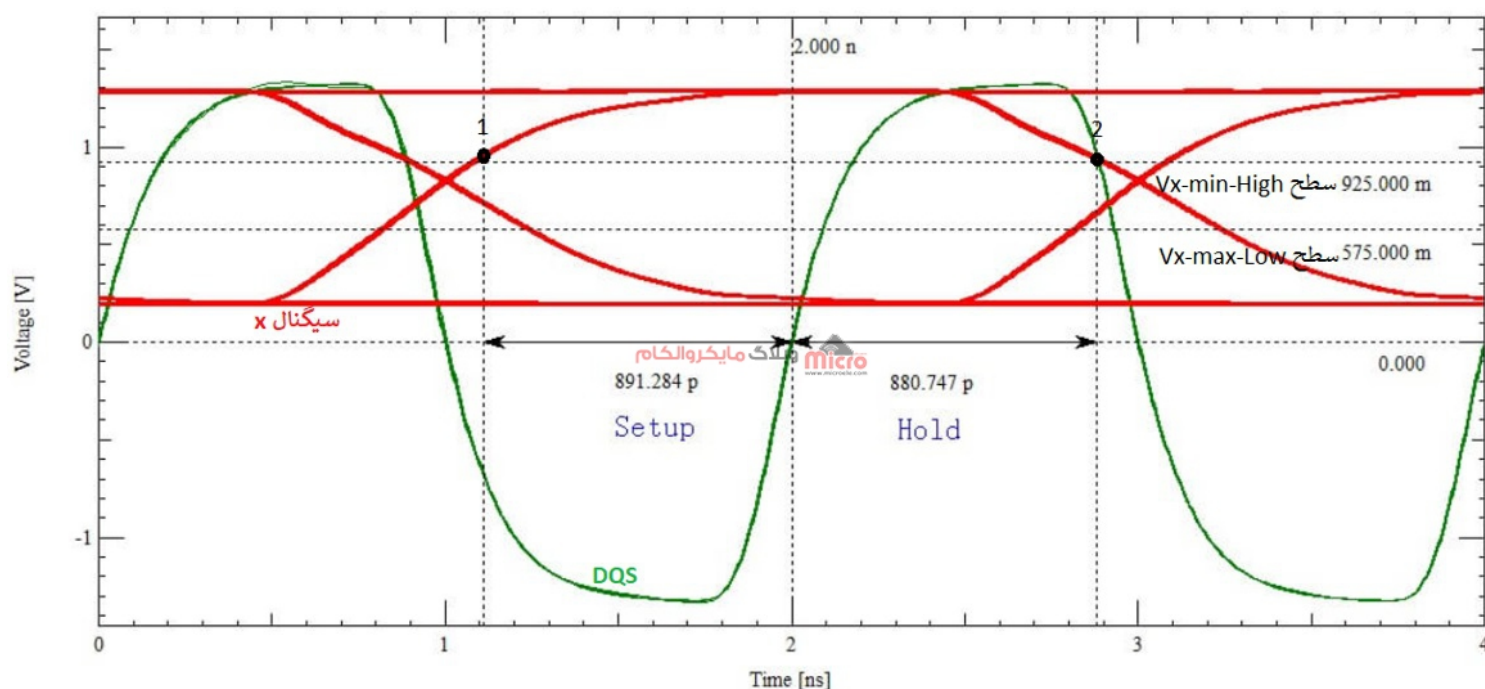
(RJ) Jitter حاصل شده را در Q Factor مربوط به BER ای که می‌خواهید به آن برسید ضرب کنید، می‌توانید محدوده جدید مربعی را برای آن BER بدست آورید.

مثال

برای $BER=10^{-20}$ مقدار Q Factor برابر با 9.2 است. اگر آن را در Random Noise (RN) و Random Jitter (RJ) ضرب کنیم، می‌توانیم محدوده مربعی جدید را برای $BER=10^{-20}$ بدست آورید. البته مشخص است که هرچه BER سختگیرانه تر شود، مربع حاصله بزرگ تر می‌شود. البته مربع مربوط به VdiVW_dV (Deterministic Noise) و Deterministic Jitter و $TdiVW_dJ$ ، یک ابعاد ثابتی دارد. که کاملاً قابل قبول است چون $VdiVW_dV$ و $TdiVW_dJ$ هر دو مقداری Deterministic هستند.

مفهوم Setup/Hold Time در DDR4

همان طور که گفتم، بین داده و کلاک آن فاصله ای وجود دارد. این فاصله باعث می‌شود همواره و تحت همه شرایط، نمونه برداری از داده در محل خاصی از داده انجام نپذیرد. اصلاً همین مشکل ممکن است باعث شود داده را به درستی نمونه برداری نکنیم. به آن زمانی که می‌توانیم کلاک DQS را جلو ببریم تا بتوانیم شروع بالا رفتن داده را پیدا کنیم (Setup Time t_{DS}) می‌گوییم. همچنین به آن زمانی که می‌توانیم کلاک DQS را نگه داریم تا دیتا برسد و بتوان محل پایین رفتن آن را پیدا کنیم (Hold Time t_{DH}) می‌گوییم. لذا t_{DS} فاصله محل بالا رفتن سیگنال با وسط DQS است و t_{DH} فاصله زمانی وسط DQS و محل پایین رفتن سیگنال است.



زمان های Setup و Hold در DDR

جمع بندی

در این قسمت از سلسله مباحث "صفر تا صد حافظه های DDR (از طراحی تا شبیه سازی)"، به بیان برخی مقدمات در باره ساختار Termination در پین های داده و کلاک و همچنین بررسی متد های شبیه سازی DDR ها پرداختیم و در نهایت، روش بهینه شبیه سازی را که همان "پاسخ ورودی و خروجی Driver و DDR ها به ورودی بیت های رندوم" است را معرفی کردیم. در بخش های آینده، بیشتر در مورد ساختار DDR ها و اصولی که در طراحی و شبیه سازی آن ها برای طراحان سخت افزار حائز اهمیت است، صحبت خواهیم کرد و بطور عملی یک مجموعه حافظه DDR را طراحی و شبیه سازی آن را دنبال می کنیم.

امیدوارم از این آموزش کمال بهره را برده باشید. در صورت داشتن هرگونه نظر یا سوال درباره این مطلب یا تجربه مشابه اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو به اشتراک بگذارید تا سایر دوستان هم بتوانند استفاده کنند. همینطور میتونید این مطلب را توی اینستاگرام با هشتگ #microelecom به اشتراک بگذارید و پیج



مایکروالکام (@microelecom) رو هم منشن کنید.