دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

4/14/2021

Homework 3

Lec 7-8

MICROPROCESSOR
AND
ASSEMBLY LANGUAGE

Spring 2021

1) دو پروتکل SPI و I2C را از نظر موارد زیر با یکدیگر مقایسه کنید.

الف) تعداد ارباب (Master)

ب) تعداد برده‌ها (Slave) و نحوه ارتباط با آن‌ها

ج) سرعت انتقال داده

د) توانایی ارسال و دریافت داده به شکل همزمان

جواب الف)

در SPI تعداد ارباب(Master) یکی میباشد و بیشتر از آن نمیشود باشد ولی در I2C میتوان هم یک ارباب و هم چندین ارباب (به صورت Multi Master) داشت.

جواب ب)

تعداد در SPI : در صورت پیاده سازی یک زنجیره از دستگاه ها به صورت موازی تعداد slave های متصل در SPI میتواند از یک تا پانزده باشد. اما اگر به صورت Daisy Chaining و درواقع به صورت متوالی پیاده سازی کنیم تعداد برده‌ها نامحدود میباشد.

نحوه ارتباط در SPI : طبق صحبت‌های استاد که باتوجه به دیتا شیت میباشد اینترفیس SPI در میکروی موردنظر ما به جای یک خط slave select ، چهارتا خط slave select دارد که میتوان به کمک آن ها به 4 تا slave به صورت مستقیم(بدون نیاز به دیکدر) وصل شد. نحوه‌ی ارتباط به این صورت است که خط MISOی ارباب به MISOی برده ها و همچنین MOSI ارباب به MOSI برده‌ها و همچنین SPCK یا همان کلاک تولید شده‌ی ارباب نیز به SPCK برده ها متصل میشود . اما برای اینکه چون همه‌ی این موارد گفته شده بین برده‌ها مشترک است برای انتخاب یک برده‌ی به خصوص از NSS مخصوص به هرکدام استفاده میشود. یعنی اگر 4 تا slave متصل باشند به هرکدام یک NSS میرود که همان Not Slave Select میباشد. هروقت ارباب روی هرکدام از این NSS ها 0 بنویسد یعنی با آن slave که این خط به آن وصل شده است کار دارد.

اما برای تعداد بیشتری slave یعنی 5 تا 15 لازم است از دیکدر 4 به 16 استفاده کنیم. که در این حالت آن خط NSS که گفتیم به عنوان ورودی های این دیکدر وصل میشوند و 16 حالت مختلف ایجاد میکنند. اما به گفته‌ی

استاد یکی از این حالت‌ها حالتی است که نمیخواهید با هیچ برده‌ای ارتباط برقرار کنید. در این حالت کانال های MISO و MOSI و SPCK به همان ترتیبی که گفتیم از ارباب به تمامی برده ها وصل میشود و دوباره کانال NSS مربوط به هر برده جداگانه به یکی از خروجی های دیکدر وصل میشوند. همچنین امکان Daisy chaining نیز وجود دارد. طبق اسلایدها لازم به ذکر است که دیتا در 8 تا 16 بیت کاراکتر منتقل میشود.

اگر بیشتر از 16 تا slave باشد چون خط سلکت نداریم مجبوریم از daisy chaining استفاده کنیم و نامحدود میتوان slave داشت.

تعداد در I2C : در روش I2C تعداد slaveهای متصل میتواند 1 تا نهایتا 128 برده باشد. (در اسلایدها آمده در برخی موارد 10 بیت آدرس هم میتوان داشت که در آن ها 1024 برده میتوان حداکثر داشت) .

نحوه‌ی ارتباط در I2C : نحوه‌ی ارتباط اینگونه است که در این پروتکل کلا دوخط SDA و SCL داریم که هم masterها (یا master)به آن وصلند هم همه‌ی slave ها. این خطوط درحالتی که هیچ دیتایی روی آن‌ها نباشد به صورت pull up هستند و 1 روی آن‌هاست. خط SCL که همان Serial Clock Line میباشد کلاکی میباشد که یکی از master ها تولید کرده‌اند(طبیعتا همه‌ی مسترها نمیتوانند همزمان روی آن کلاک دلخواه خودشون را بذارن و یکی میذارد) و بقیه‌ی کارها از قبیل تعیین کردن slave موردنظر و نوع ارتباط(Read/Write) و دیتای ارسالی روی خط SDA مشخص میشود. ارتباط به این صورت است که یکی از master ها که میخواهد ارتباطی برقرار کند ابتدا یک بیت start که 0 هست به اندازه یک کلاک روی خط میگذارد و تمامی slaveها آماده میشوند که از بیت بعد به اندازه 7 بیت آدرس را بگیرند در نهایت یکی از برده ها انتخاب میشود و از آنجا به بعد فقط با همان کار داریم. سپس در بیت بعد نوع ارتباط(Read/Write) مشخص میشود. سپس برده یک بیت ACK میدهد تا ارتباط برقرار شود. سپس یک بایت یک بایت دیتا منتقل میشود و انتهای هرکدام یک بیت ACK فرستاده میشود که اگر درحال نوشتن باشیم این بیت را برده میگوید و اگر مستر درحال خواندن از برده باشد بیت ACK را Master میفرستد. در انتها Master یک بیت stop میفرستد که یعنی ارتباط تمام شده است. (همچنین آدرس و دیتای ارسالی از پرارزش ترین بیت خود ارسال میشوند)

جواب ج)

SPI : طبق اسلایدها(صفحه 5 لکچر 7) سرعت انتقال داده‌ها چند مگابایت بر ثانیه میباشد.

(Multiple Mbps transmission speed)

I2C : طبق اسلایدها(صفحه5 لکچر 8) چندین مود برای تعیین سرعت انتقال داده میتواند وجود داشته باشد:

Standard mode : 100 kbit/s

Fast mode : 400 kbit/s

Low-speed mode : 10 kbit/s

Fast mode plus : 1Mbit/s

High speed mode : 3.4 Mbit/s

Ultra Fast-mode : 5 Mbit/s

(البته طبق صفحه 10 لکچر 8 در میکروی ما در سرعت انتقال I2C میتواند در بیشینه حالت 400 Kbits/s باشد.)

درکل میتوان گفت که SPI سریع تره و I2C میتواند سرعتی در حد SPI یا کمتر از آن داشته باشد(طبق صحبت های استاد در اسلایدها)

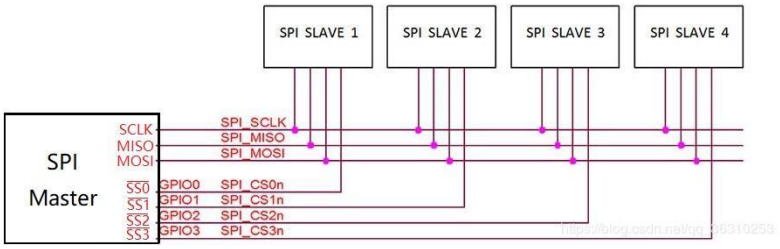
جواب د)

SPI یک پروتکل Full Duplex میباشد یعنی توانایی ارسال و دریافت داده به صورت همزمان را دارد. چون دو سیم مجزا برای انتقال دیتا وجود دارد.

I2C یک پروتکل Half Duplex میباشد یعنی در هر لحظه برای مثال Master میتواند یا فقط دیتا بگیرد یا فقط دیتا بفرستد چون یک سیم انتقال دیتا بیشتر نداریم.

2) برای پیاده‌سازی یک زنجیره از دستگاه‌ها به کمک رابط SPI، دو راه سری و موازی وجود دارد. این دو روش را مقایسه کنید و مزایا و معایب هرکدام را ذکر کنید.

در حالت موازی که به صورت زیر میباشد:



در این رویکرد Master به کمک یک NSS منحصر به فرد برای هر slave میتواند آن را انتخاب کند. برده هایی که انتخاب نشده اند خروجی یا data out آن ها به حالت های امپدانس میرود و اون برده ای که انتخاب شده شده در اصل دیتا به MISOی میکرو میدهد. همچنین آنslave ای که انتخاب شده نیز میتواند از روی خط MOSI اطلاعات بردارد.

در این حالت slave های ما میتوانند رویکردها و روش های مختلفی در زمینه کلاک (clock edge یا idle clock state) داشته باشند و به صورت مستقل قابل تنظیم اند.

مزیت آن این است که :

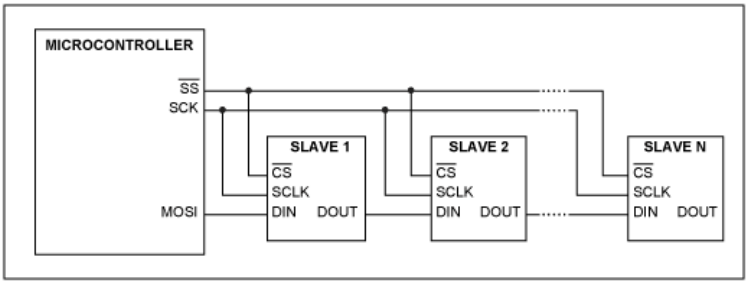
از نظر نرم افزاری کار زیادی نیاز نداریم.

همچنین در این حالت میزان تاخیر کم تر است و زمان ارسال دیتا به هریک از slave ها یکسان است.

از معایب آن این است که :

به ازای هر slave ما نیاز به یک chip select منحصر به فرد داریم. (در میکروی ما برای 4 تا slave میتوان به صورت مستقیم پین وصل کرد یا اگر بیشتر از 4 تا شد به کمک دیکدر سلکت slave ها را به هر یک از خروجی‌های دیکدر وصل کرد). در واقع در اینجا محدودیت بیشتری روی تعداد slave ها داریم.

در روش سری که از Daisy Chaining استفاده میشود در واقع به صورت زیر میباشد:



در این حالت slave ها به صورت cascade شده هستند و خروجی یکی ورودی دیگری میباشد. در این حالت با یک NSS در Master کارمان راه میوفتد. ابتدا میکرو به slave اول به صورت مستقیم دیتا را میدهد و در شیفت رجیستر داخلی آن میرود سپس تا زمانی که روی NSS منطق 0 باشد و کلاک فعال باشد این دیتا به همین ترتیب از هر برده به بده‌ی بعدی میرود و به همین ترتیب propagate میشود تا هر slave دستور مناسب خود را دریافت کند.

در این حالت تمامی slave ها باید از clock edge و clock idle state یکسان استفاده کنند.

مزیت آن این است که:

در این حالت یک chip select یکسان برای slave ها استفاده میکنیم در نتیجه تعداد slave های بیشتری میتوانیم داشته باشیم.

همچنین از پین های کمتری از میکرومون استفاده میکنیم و اگر Master ما فقط یک NSS هم داشته باشد میتوانیم چندین slave را به آن وصل کنیم.

از معایب آن این است که:

از لحاظ نرم افزاری باید کار بیشتری انجام دهیم چرا که باید مطمئن شویم تعداد بیت‌های صحیح از slave صحیح دریافت و ارسال شده است.

در این جا چون slave ها به صورت سری بسته شدند زمان دسترسی و ارسال دیتا به آخرین slave با اولین slave برابر نیست و این زمان افزایش میابد.

3) اگر در یک دستگاه ارباب شمار پایه های CS/NSS کمتر از شمار دستگاه های برده باشد، آنگاه چه راه هایی برای وصل کردن و ارتباط با دستگاه های برده از راه آن باس SPI پیشنهاد میکنید؟

همانطور که در سوال قبل اشاره شد دو راه میتوان استفاده کرد:

1- میتوان از دیکدر برای ارتباط با تعداد بیشتری slave استفاده کرد. البته در این حالت باید باز تعداد slave ها از تعداد خروجی‌های دیکدر کمتر باشد. در میکروی ما که 4 تا پایه‌ی NSS برای یک Master وجود دارد میتوان از یک دیکدر 4 به 16 استفاده کرد و تا 16 تا slave را ساپورت کرد. هر یک از خروجی های دیکدر به NSS یکی از slave ها وصل میشود و هر زمان که خواستیم میتوانیم یکی از آن ها را فعال کنیم که از روی MOSI دیتا بردارد یا روی MISO دیتا بگذارد.

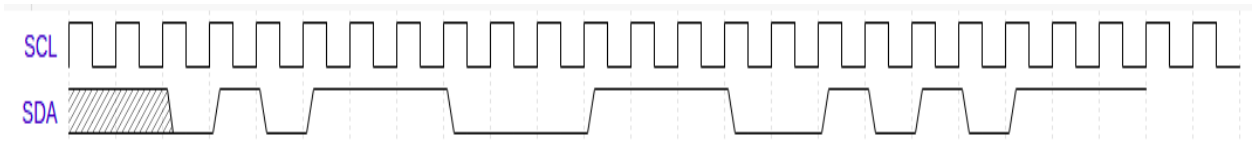
2- همچنین میتوان از قابلیت Daisy Chaining استفاده کرد. یعنی به صورت سری slave ها را به هم وصل کنیم و از یک NSS برای فعال کردن آن ها استفاده کنیم. در این حالت دیگر Master با مثلا برده‌ی چهارم ارتباط مستقیم ندارد اما میتواند پیام را به آن برساند. در این حالت slave ها به صورت cascade شده هستند و خروجی یکی ورودی دیگری میباشد. ابتدا میکرو به slave اول به صورت مستقیم دیتا را میدهد و در شیفت رجیستر داخلی آن میرود سپس تا زمانی که روی NSS منطق 0 باشد و کلاک فعال باشد این دیتا به همین ترتیب از هر برده به بده‌ی بعدی میرود و به همین ترتیب propagate میشود تا هر slave دستور مناسب خود را دریافت کند.

4) شکل موج زیر مربوط به پروتکل I2C می‌باشد. SDA در ابتدا در حالت بیکار (Idle) بوده است. با توجه به داده ارسالی موارد زیر را مشخص کنید.

الف) داده‌ی ارسالی

ب) آدرس برده

ج) نوع عملیات مدنظر(خواندن/نوشتن)

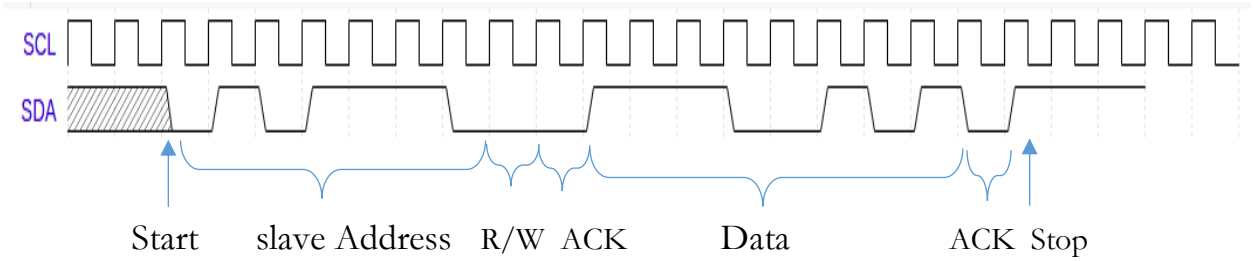


طبق اسلاید ها شکل کلی انتقال داده در I2C به صورت زیر میباشد:

High to Low 7bit 1bit 1 bit 8 bit 1 bit Low to High

| start | slave Address | R/W | ACK | Data | ACK | stop |

حال بنابراین داریم:



با 1 به 0 شدن SDA در لبه‌ی High از SCL ، Master ارتباط را شروع میکند.

الف) داده‌ی ارسالی (Data) : 11100101

بیت دهم تا هفدهم (به اندازه 8بیت) همانطور که در شکل نیز مشخص شده است داده‌ی ارسالی یا دیتای ماست.

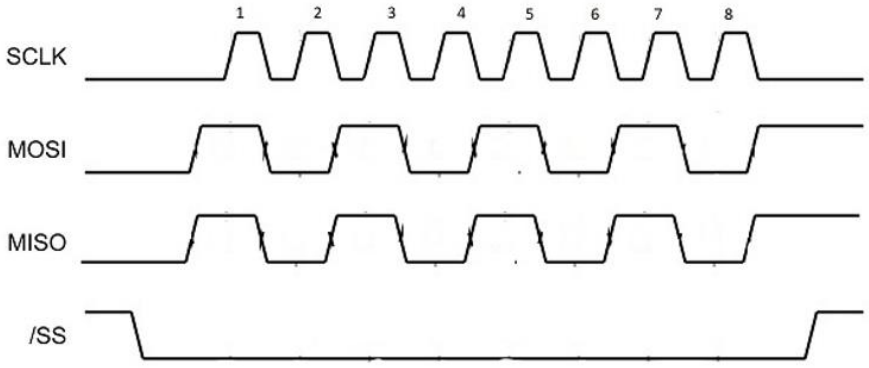
ب) آدرس برده (Slave Address) : 0101110

بیت اول تا هفتم (به اندازه 7 بیت) همانطور که در شکل نیز مشخص شده است آدرس برده میباشد.

ج) نوع عملیات مورد نظر (خواندن/ نوشتن): بیت مربوط به آن 0 میباشد یعنی عملیات Write میباشد. و درواقع master روی slave مورد نظر می نویسد.

بیت هشتم نشان دهنده‌ی نوع عملیات(read/write) میباشد.

5) در شکل موج زیر که توسط یک رابط SPI روی باس قرار می‌گیرد، زمان شروع و پایان ارسال داده، محتوا داده ارسالی و نوع هر عملیات (نوشتن روی ارباب یا برده) را مشخص کنید.



در رابط SPI نشان داده شده با توجه به اینکه slave select مربوطه Active Low میباشد زمانی که روی آن منطق 0 باشد و کلاک هم برقرار باشد ارتباط با برده‌ی مورد نظر برقرار شده و ارسال داده ها شروع میشود و زمانی که کلاک قطع شود یا روی NSS مقدار 1 نوشته شود ارسال داده به اتمام میرسد.

بنابراین در کلاک1 (به طور دقیق تر از لبه‌ی بالا رونده‌ی کلاک 1) ارسال داده شروع میشود و در کلاک 8 (به طور دقیق‌تر در لبه‌ی پایین رونده‌ی کلاک 8) به پایان میرسد .

داده‌ای که slave دریافت کرده و Master فرستاده:(از روی دیتاهای روی خط MOSI مشخص میشود):

10101010

در این حالت Master روی Slave دیتا مینویسد.

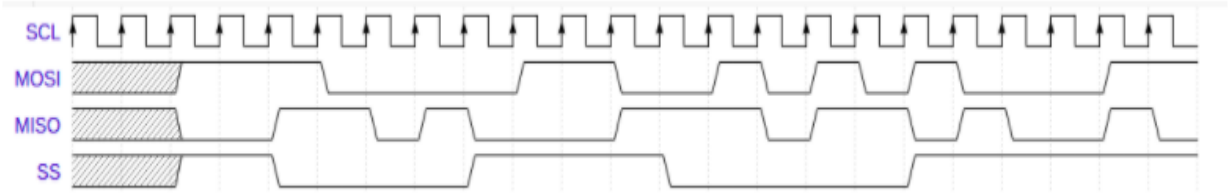
داده ای Master فرستاده و Slave دریافت کرده (از روی دیتاهای روی خط MISO مشخص میشود):

10101010

در این حالت Slave روی Master دیتا مینویسد.

در این حالت هم Slave و هم Master عمل Read و Write را انجام میدهند. عمل شیفت در لبه‌ی بالارونده و عمل capture در لبه‌ی پایین رونده انجام میشود.

6) شکل زیر مربوط به ارتباط از طریق پروتکل SPI می‌باشد. زمان‌های انتقال داده و همچنین داده‌ی منتقل شده را تعیین کنید.



همانطور که در مثال قبل هم گفتیم باید هردو شرط انتخاب شدن برده و برقراری کلاک اوکی باشند تا داده انتقال یابد.

در اینجا نیز هرجا SS ما 0 باشد داده منتقل میشود و درواقع Active Low میباشد . پس زمانی که کلاک برقرار باشد و منطق روی خط SS برابر 0 باشد در این حالت برده‌ی موردنظر انتخاب شده و دیتا منتقل میشود.

بنابراین یک بار در کلاک پنجم (به طور دقیق تر لبه‌ی بالا رونده‌ی کلاک پنجم) تا کلاک هشتم (به طور دقیق تر لبه‌ی پایین رونده کلاک هشتم) داده منتقل میشود. این دیتا به صورت زیر میباشد:

دیتایی که slave دریافت کرده و Master فرستاده:(از روی دیتاهای روی خط MOSI مشخص میشود):

1000

دیتایی که Master دریافت کرده و Slave فرستاده (از روی دیتاهای روی خط MISO مشخص میشود):

1101

و بار دیگر درکلاک سیزدهم (یه طور دقیق‌تر درلبه‌ی بالا رونده‌ی کلاک سیزدهم) تا کلاک هفدهم(به طور دقیق‌تر لبه‌ی پایین رونده‌ی کلاک هفدهم) انتقال دیتا انجام میشود.این دیتا به صورت زیر میباشد:

دیتایی که slave دریافت کرده و Master فرستاده:(از روی دیتاهای روی خط MOSI مشخص میشود):

01010

دیتایی که Master دریافت کرده و Slave فرستاده (از روی دیتاهای روی خط MISO مشخص میشود):

11011

7)

أ. پروتکل SPI دارای چند سیم ارتباطی می‌باشد؟ عملکرد هر سیم را به اختصار توضیح دهید.

ب. بهترین نرخ بازدهی را که با پیکربندی هر یک از پروتکل ها UART, SPI, TWI به ازای فرستادن n بایت داده می توان به آن رسید، به دست آوردید. (#بیت فریم / #بیت داده)

أ) پروتکل SPI در حالت کلی 4 سیم ارتباطی دارد و همچنین به عنوان 4wire bus نیز شناخته شده است.

MOSI : (Master Out Slave In) : یکی از خطوط انتقال دیتاست. این سیم وظیفه‌ی انتقال داده‌های خروجی از Master به ورودی Slave(ها) را دارد. به عبارتی اگر master بخواهد داده‌ای به slave (ها) بفرستد روی این سیم این کار را انجام میدهد.

MISO: (Master In Slave Out): یکی از خطوط انتقال دیتاست. این سیم وظیفه‌ی انتقال داده‌های خروجی از یک slave به ورودی Master را دارد. به عبارتی اگر Slave بخواهد داده‌ای به Master ارسال کند، روی این سیم این کار را انجام میدهد.

SPCK (Serial Clock) : یک خط کنترلی میباشد. توسط master تولید می‌شود و جریان بیت های داده را تنظیم می کند. master ممکن است داده ها را با baud rateهای مختلف انتقال دهد. در خط SPCK به ازای انتقال هر بیت یک cycle یا یک تناوب رخ میدهد یا هربیت با یک cycle منتقل میشود.

NSS (Slave Select): یک خط کنترلی میباشد. به کمک آن میتوان یک slave را فعال یا غیر فعال کرد. در اینجا یک خط Active Low میباشد و از سمت Master فعال میشود. وقتی کلاک برقرار باشد از زمانی که Master این خط را فعال کند ارتباط با slave موردنظر که این خط به NSS آن نیز وصل است ، برقرار میشود.

(البته در میکروی ما برای مثال 4 تا NSS وجود دارد ولی خب همگی شبیه هم هستن و کار متمایزی ندارن بلکه برای ساپورت کردن slave های بیشتری هستند)

ب) در TWI فرمت فریم به صورت زیر است:

High to Low 7bit 1bit 1 bit 8 bit 1 bit Low to High

| start | slave Address | R/W | ACK| Data | ACK | stop|

به ازای هر بایت دیتا یک بیت ACK نیز اضافه میشود. (بیت های استارت و استاپ در نظر گرفته نشدن درواقع آن ها را یک transition میدانیم مطابق حرف یکی از تی ای ها) بنابراین تعداد بیت فریم آن برابر است با :
9+9n

هم چنین تعداد بیت داده‌ی آن نیز برابر است با: 8n

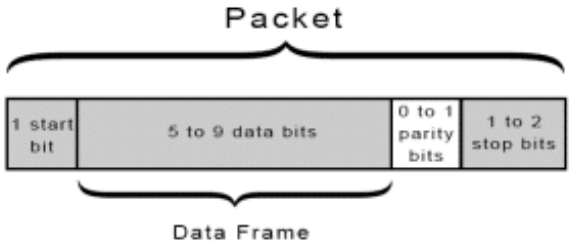
بنابراین نرخ بازدهی برابر است با: $\frac{8n}{9+9n}$ این نرخ به ازای افزایش n افزایش میابد. بنابراین با گرفتن حد آن به ازای nهای بزرگ بهترین نرخ بازدهی برابر با $\frac{8}{9}$ میشود یعنی : 88.8 %

(البته اگه استارت و استاپ رو به عنوان دو بیت در کل فریم در نظر بگیریم آنگاه به $\frac{8n}{11+9n}$ میرسیم بازهم بهترین نرخ حدودا همان 88.8% میشود.)

در SPI تعداد بیت فریم با تعداد بیت داده برابر است و بیت‌های ACK یا Start یا Stop یا Address ندارد.
بنابراین بهترین نرخ بازدهی آن برابر است با$\frac{8n}{8n}$ یا درواقع: 100%

در UART طبق شکل صفحه 6 لکچر 9 بیت parity میتواند 0 بیت یا یک بیت باشد. هم چنین اگر بیت استارت و استاپ را هم در نظر نگیریم (طبق سوالی که از یکی از تی ای ها پرسیدم) برای رسیدن به بهترین نرخ بازدهی باید parity را 0 بیت و داده را 9 بیت در نظر بگیریم . پس در این حالت ما فقط داده را ارسال میکنیم در packet و بهترین نرخ بازدهی برابر است با $\frac{9n}{9n}$ یا درواقع: 100%

البته اگر میخواستیم بیت استارت و استاپ رو هم در نظر بگیریم به $\frac{9n}{9n+2}$ میرسیدیم که خیلی به بازده 100 درصد نزدیک میباشد.



هم چنین در این مورد آخر بنده دیتاها را 9 بیتی گرفتم. اگر بخواهیم طبق صورت سوال که گفته n بایت پس داده‌ها را باید 8 بیتی در نظر بگیریم ولی باز هم تغییر چندانی حاصل نمیشود و نرخ بازده در حالتی که بیت استارت

و استاپ را در نظر نگیریم(درواقع یک transition باشد) میشود $\frac{8n}{8n}$ که همان 100 درصد میشود. اگر هم بیت استارت و استاپ در نظر بگیریم میشود $\frac{8n}{8n+2}$ که باز هم نزدیک به 100 میباشد.

- مهلت ارسال تمرین ساعت 23:59 روز جمعه 10 ام اردیبهشت می‌باشد.
- سوالات خود را می‌توانید تنها از طریق ایمیل AUTMicroTA@gmail.com بپرسید.
- ارائه پاسخ تمرین بهتر است به روش های زیر باشد:
  1) استفاده از فایل docx. تایپ پاسخ‌ها و ارائه فایل Pdf
  2) چاپ تمرین و پاسخ دهی به صورت دستنویس خوانا
- فایل پاسخ تمرین را تنها با قالب **HW2 - 9731***.pdf** در مودل بارگزاری کنید.
  - نمونه: HW2- 9731747
- فایل زیپ ارسال **نکنید**.