

به نام خدا

آزمایشگاه ریزپردازنده دانشکده مهندسی کامپیوتر

آزمایش شماره یازده

آشنایی با استاندارد I2C و کار با آن

همانطور که می‌دانید اساسی‌ترین روش برای ارتباط بین پردازنده و اجزاء جانبی، استفاده از یک باس داده و خطوط آدرس موازی است. با استفاده از این روش، علاوه بر اینکه فضای زیادی از برد مدار چاپی صرف انتقال خطوط آدرس و داده می‌شود به تعدادی دیکودر آدرس و مدارات منطقی برای اتصال وسایل نیاز خواهد بود.

در وسایلی مانند تلویزیون، ویدئو و تجهیزات صوتی که به تولید انبوه می‌رسند صرف این هزینه، معمولاً قابل قبول نخواهد بود. در این کاربردها، هر قطعه‌ای که صرفه‌جویی شود در قیمت محصول موثر بوده و این مسئله به سود تولید کننده خواهد بود. علاوه بر این، تعداد خطوط کنترلی و قطعات بیشتر به معنای حساسیت بیشتر سیستم در برابر EMI، تست و عیب‌یابی مشکل‌تر و توان مصرفی بالاتر است.

نتیجه‌ی تحقیقاتی که در اوایل دهه‌ی ۱۹۸۰ شرکت فیلیپس برای غلبه بر این مشکلات انجام داد باس دو سیمه‌ای بود که I^2C (IIC) نامیده شد I2C سرنام Inter IC است که به طور ضمنی گویای عملکرد آن است: فراهم کردن یک مسیر ارتباطی بین مدارات مجتمع.

امروزه این استاندارد به طور عمومی در صنعت الکترونیک پذیرفته شده و کاربرد آن از تجهیزات صوتی و تصویری فراتر رفته است. در حال حاضر باس I^2C بوسیله‌ی بیش از ۵۰ تولید کننده نظیر DT Microelectronics، Intel، Infineon، TI، Maxim، Atmel، Analog DeVICES و غیره پذیرفته شده است و در بیش از هزاران مدار مجتمع مختلف از آن استفاده می‌شود. به عنوان نمونه سه حوزه از کاربردهایی که در آن‌ها از باس I^2C به عنوان یک مسیر ارتباطی استفاده می‌شود، عبارتند از:

- کنترل کننده‌های هوشمند که معمولاً به صورت میکروکنترلرهای تک چیپ هستند.
- مدارات مجتمع با کاربردهای عمومی مانند: درایور LCD، Remote I/O Port، حافظه‌ها (مانند SRAM، VNRAM و EEPROM)، مبدل‌های داده (مانند ADC) و

- مدارات با کاربرد خاص مانند: تنظیم کننده‌های دیجیتال و پردازنده‌های سیگنال در سیستم-های رادیویی و ویدئویی، تولید کننده‌ی DTMF، انواع حسگرها، Real-Time Clock و

...

آشنایی با نسخه‌های استاندارد I²C

I²C یک استاندارد سریال و همزمان و Half-Duplex است، بنابراین نیاز به دو خط کلاک و داده می‌باشد. این استاندارد به صورت سخت‌افزاری امکان راه‌اندازی یک شبکه از قطعات الکترونیکی را تنها با دو مقاومت Pull-up فراهم می‌کند. سرعت ارتباط، حداکثر وسایل مجاز و برخی از مشخصات دیگر باس، بستگی به نسخه‌ی استاندارد دارد که در ادامه به بررسی مختصر سرعت و عرض آدرس در هر یک از نگارش‌های I²C می‌پردازیم.

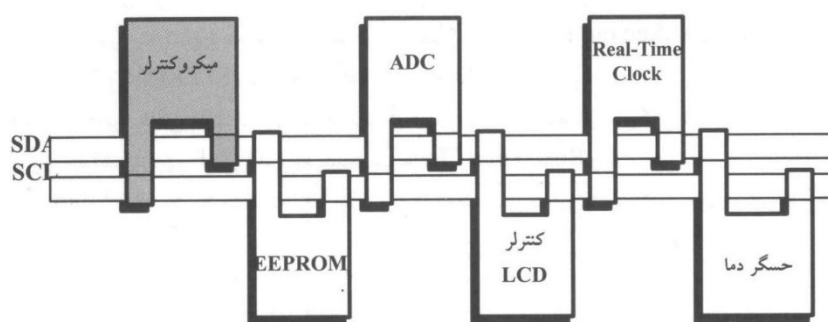
- نسخه‌ی اولیه استاندارد I²C (سال ۱۹۸۲) تنها از آدرس ۷ بیتی با حداکثر سرعت ۱۰۰ کیلوبیت بر ثانیه (Standard Mode) پشتیبانی می‌کرد.
- در نسخه‌ی دوم که در سال ۱۹۹۲ ارائه شد سرعت ارتباط تا ۴۰۰ کیلوبیت در ثانیه (Fast Mode) افزایش یافته و به علت ۱۰ بیتی شدن آدرس، تعداد وسایل مجاز به ۱۰۲۴ افزایش یافت.
- نسخه‌ی بعدی در سال ۱۹۹۸ ارائه شد. در این نسخه، حداکثر سرعت ۳,۴ مگابیت بر ثانیه (High Speed Mode) بوده و آدرس همچنان ده بیتی است.
- آخرین نگارش استاندارد I²C در سال ۲۰۰۰ انتشار یافت و در آن، سرعت و عرض آدرس نسبت به نسخه‌ی ۲ تغییری نکرده است.

آشنایی با مفاهیم باس I²C

در باس I²C داده به کمک دو خط SDA و SCL منتقل می‌شود (کلمه SDA مخفف Serial Data و کلمه SCL مخفف Serial Clock می‌باشند). هر وسیله‌ای خواه میکروکنترلر باشد یا یک حسگر یا حافظه و یا قطعه‌ای دیگر) با یک آدرس منحصر به فرد شناخته می‌شود و می‌تواند به صورت فرستنده (Transmitter) و یا گیرنده (Reciever) عمل کند. از نقطه نظر دیگر، هر وسیله می‌تواند Master یا Slave باشد. Master وسیله‌ای است که انتقال داده را آغاز کرده، کلاک را تولید می‌کند و انتقال را خاتمه می‌دهد. Slave وسیله‌ای است که بوسیله Master خطاب شده است (آدرس‌دهی گردیده است).

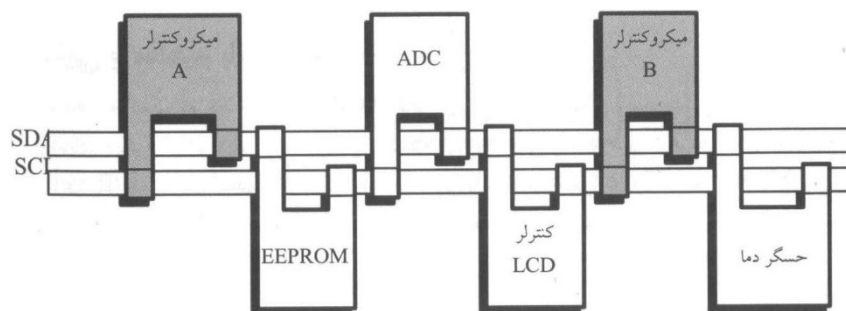
در ساده‌ترین حالت، باس I2C شامل یک Master و یک Slave است. Master که معمولاً یک وسیله‌ی قابل برنامه‌ریزی (مانند میکروپروسسور یا میکروکنترلر) است می‌توان ارتباط را با Slave آغاز کرده، آن را کنترل کند و در نهایت پایان دهد.

شکل زیر را در نظر بگیرید، این باس I2C که شامل یک آرایش پیچیده‌تری است شامل یک میکروکنترلر به عنوان Master و چندین وسیله‌ی Slave است. فرض کنید که میکروکنترلر می‌خواهد داده‌ای را برای نمایش به کنترلر LCD ارسال نماید. در این شرایط، میکروکنترلر در وضعیت Master-Transmitter قرار می‌گیرد و کنترلر LCD، در نقش Slave-Receiver خواهد بود. میکروکنترلر وظیفه دارد تا با آدرس‌دهی، Slave مورد نظر (کنترلر LCD) را انتخاب کرده و داده را برای آن ارسال نماید. پس از اتمام ارسال، ارتباط بین Master و Slave توسط Master پایان می‌یابد.



حال وضعیتی را در نظر بگیرید که میکروکنترلر می‌خواهد داده‌ای را از حسگر دما دریافت کند، در این حالت، میکروکنترلر در نقش Master-Receiver و حسگر دما به عنوان Slave-Transmitter است. در اینجا نیز میکروکنترلر وظیفه دارد تا با آدرس‌دهی، Slave مورد نظر (حسگر دما) را انتخاب کرده و داده را از آن دریافت نماید. پس از اتمام دریافت، ارتباط Master و Slave توسط Master پایان می‌یابد.

در استاندارد I2C، این امکان وجود دارد تا بیش از یک وسیله که توانایی کنترل باس را دارند به باس متصل شده باشند. به این دلیل، این باس یک باس Multi-Master می‌باشد. شکل زیر وضعیتی را نشان می‌دهد که دو میکروکنترلر و چهار وسیله دیگر بر روی باس حاضرند. در صورتی که میکروکنترلر A یا B بخواهد با هر یک از چهار وسیله مختلف ارتباط برقرار کند، آن میکروکنترلر (مانند وضعیت قبل) در نقش Master-Transmitter یا Master-Receiver بوده و وسیله‌ی دیگر نیز Slave-Receiver یا Slave-Transmitter خواهد بود.

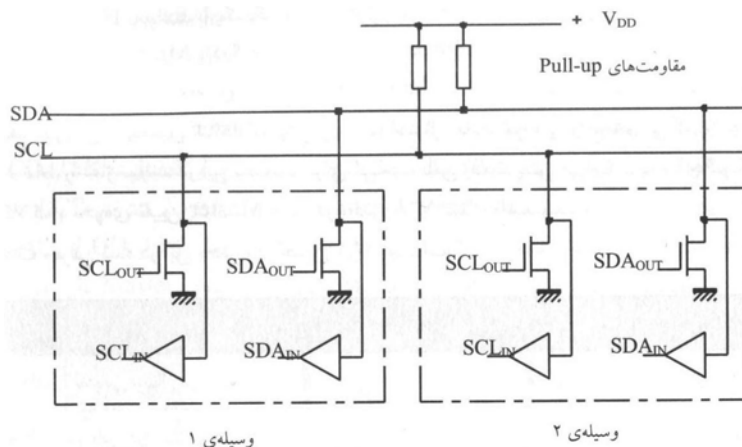


حال اگر دو میکروکنترلر A و B بخواهند با یکدیگر ارتباط برقرار نمایند لازم است تا یکی از آنها در نقش Master-Transmitter یا Master-Receiver و دیگری Slave-Transmitter یا Slave-Receiver باشد.

در تمام حالت‌های اشاره شده، در هر لحظه تنها یک Master و Slave با یکدیگر ارتباط دارند. اما وجود دو میکروکنترلر بر روی باس (که هر دوی آنها پتانسیل Master بودن را دارند) احتمال ایجاد شرایطی را فراهم می‌کند تا دو Master به طور همزمان یک ارتباط را آغاز نمایند. در این وضعیت، برای تصاحب باس رقابت پیش می‌آید که با چگونگی همزمان‌سازی پالس‌های کلاک و نحوه‌ی تعیین Master آشنا خواهید شد.

آشنایی با مشخصات فیزیکی باس

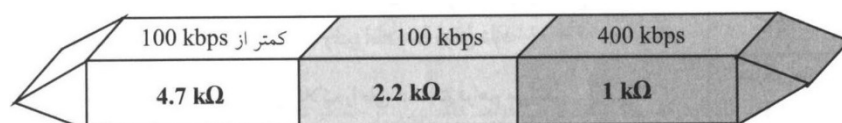
همانطور که اشاره شد، خطوط ارتباطی باس I²C شامل SDA و SCL است که به ترتیب حامل کلاک و داده می‌باشند. SDA و SCL خطوط دو طرفه‌ای هستند که هر یک بوسیله‌ی یک مقاومت Pull-up به پایانه‌ی مثبت تغذیه (VDD یا VCC) وصل شده‌اند. (مطابق با شکل زیر)



مطابق با استاندارد، زمانی که باس آزاد است و هیچ انتقالی بر روی آن انجام نمی‌شود، باید هر دو خط SDA و SCL یک منطقی باشند. بنابراین طبقه‌ی خروجی قطعات متصل شده به باس I²C به صورت Open-collector یا Open-drain طراحی می‌شوند. در حالت بیکاری با خاموش شدن ترانزیستور، طبقه‌ی خروجی از خط جدا شده و مقاومت Pull-up آن را در سطح یک نگاه می‌دارد.

با توجه به این مسئله دو سطح منطقی Float-High و Drain-Low بر روی باس تعریف می‌شود. Float-High وضعیتی است برای ایجاد یک منطقی، که در آن تمام ترانزیستورها خاموش بوده و خروجی آنها شناور است. همان طور که از نام Drain-Low برمی‌آید تاثیر یک وسیله بر روی خط با صفر منطقی است. با توجه به این ویژگی، می‌توان گفت باس I²C یک خط Wired-AND یا AND سیمی است، بدین ترتیب که اگر تنها یک سیگنال صفر بر روی خط ارسال شود، سطح آن صفر می‌شود (این رفتار مشابه با رفتار گیت AND منطقی است که اگر تنها یک ورودی آن صفر شود خروجی صفر خواهد داشت).

با توجه به توضیحات بالا، نقش مقاومت Pull-up کاملاً تعیین کننده است. مقدار این مقاومت متناسب با سرعت انتقال تعیین می‌شود. در سرعت‌های پایین، مقدار تقریبی ۴٫۷ کیلو اهم و در سرعت‌های نزدیک به ۴۰۰ کیلو بیت بر ثانیه، یک کیلو اهم می‌تواند مناسب باشد. (به شکل زیر دقت کنید)



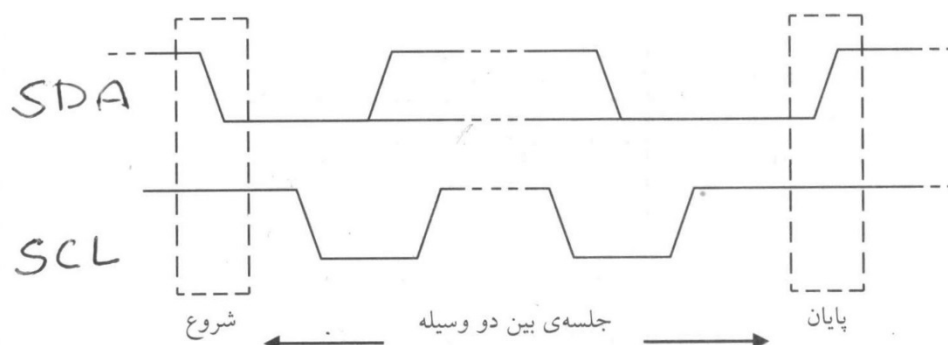
آشنایی با عناصر I²C

داده‌ای که بر روی باس I²C منتقل می‌شود قابل تفکیک به عناصر شرایط شروع (Start Condition)، شرایط پایان (Stop Condition)، شرایط بازشروع (Restart Condition)، انتقال داده (Data Transfer)، شرایط تصدیق (Acknowledge Condition) و شرایط عدم تصدیق (Not Acknowledge Condition) است که در ادامه به بررسی هر یک می‌پردازیم:

• شرایط شروع و پایان

در استاندارد I²C، برای برقراری ارتباط بین Master و Slave لازم است تا با ایجاد شرایط شروع توسط Master، جلسه‌ای (session) بین Master و Slave برقرار شود. همانطور که اشاره شد در حالت

بی‌کاری، هر دو خط SCL و SDA در وضعیت یک منطقی (Float-High) می‌باشند. برای ایجاد وضعیت شروع کافی است تا در حالی که SCL همچنان یک است خط SDA توسط Master صفر شود. (شکل زیر)



پس از انتقال داده بین Master و Slave به منظور ختم جلسه‌ی برقرار شده، Master یک شرایط پایان بر روی باس ایجاد می‌کند. در وضعیت پایان در حالیکه خط SCL یک منطقی است، SDA از صفر به یک تغییر می‌کند (رها می‌شود).

در فاصله بین شرایط شروع و پایان، باس مشغول بوده و استفاده از آن توسط سایر وسایل امکان پذیر نمی‌باشد.

• شرایط باز شروع

همانطور که اشاره شد، به منظور شروع ارتباط بین Master و Slave یک وضعیت شروع توسط Master ایجاد می‌شود. پس از ایجاد وضعیت شروع، با ارسال یک یا چند بایت داده، Slave مورد نظر و برخی مشخصه‌های ارتباط را تنظیم می‌کند. از آنجایی که اعمال این مشخصات تنها در امتداد وضعیت شروع قابل اعمال است در صورتی که Master قصد تغییر یک یا چند مورد از آنها را داشته باشد می‌تواند بدون ایجاد وضعیت پایان، مجدداً یک وضعیتی شروع ایجاد نماید. به این وضعیت، باز شروع گفته می‌شود.

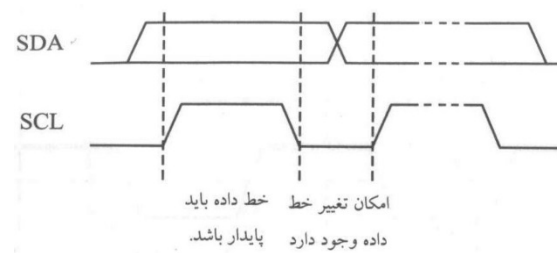
به منظور قیاس، یک ارتباط تلفنی را در نظر بگیرید. برای شروع ارتباط لازم است تا ابتدا گوشی تلفن بوسیله‌ی شما برداشته شود (وضعیت شروع)، پس از آن می‌توانید شماره‌گیری کرده (تنظیم مشخصه‌های ارتباط) و صحبت نمایید (مبادله داده)، در نهایت معمولاً با گذاشتن گوشی، ارتباط را قطع می‌کنید (شرایط پایان). بدیهی است که در فاصله‌ی بین برداشت گوشی (وضعیت شروع) تا گذاشتن آن (وضعیت پایان) خط مشغول بوده و برابری ارتباط توسط شخص (Master) دیگری امکان پذیر نمی‌باشد.

حال اگر پس از اتمام ارتباط فوق، قصد داشت باشید تا ارتباطی دیگر برقرار کنید، می‌توانید بدون گذاشتن گوشی تلفن (شرایط پایان) و تنها با یک بار فشردن قلاب، ارتباط دیگری را آغاز کرده (وضعیت باز شروع) و شماره‌گیری نمایید (تنظیم مشخصه‌های ارتباط).

• انتقال داده

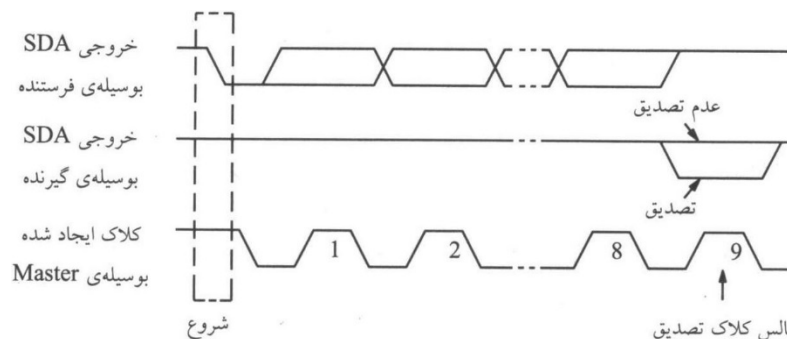
در فاصله‌ی بین شروع و پایان، بیت‌های داده منتقل می‌شوند. هر بیت داده‌ی ارسال شده در لبه بالا رونده پالس کلاک خوانده می‌شود.

سطح منطقی خط SDA تنها در وضعیتی می‌تواند تغییر کند که SCL در سطح صفر است (شکل زیر) به عبارت دیگر، طی مدتی که خط SCL یک است باید داده‌ی موجود بر روی خط SDA پایدار بماند در غیر- اینصورت تغییر ایجاد شده، شرایط آغاز یا پایان تلقی می‌شود.



• شرایط تصدیق و عدم تصدیق

I²C یک استاندارد هشت بیتی است و بنابراین داده به صورت بایت به بایت ارسال می‌شود. در این استاندارد لازم است تا گیرنده پس از دریافت هر بایت، برای فرستنده تصدیق ارسال کند تا فرستنده از سرنوشت بایت ارسال شده، آگاهی یابد.

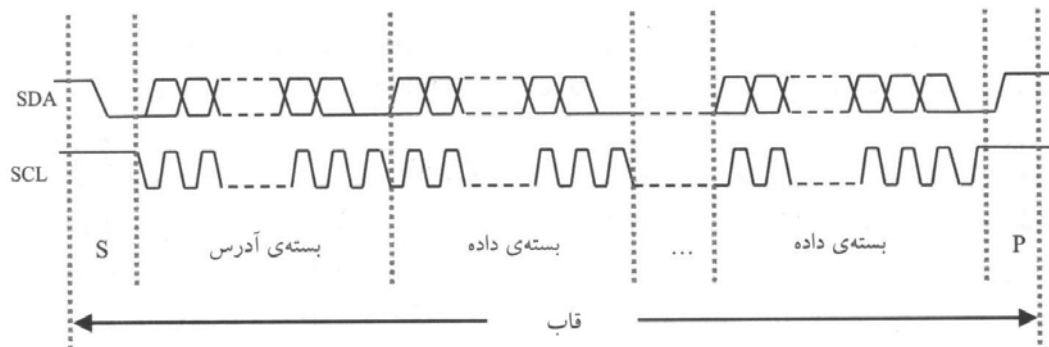


مطابق با شکل فوق در سیکل کلاک نهم، فرستنده پایه SDA را رها کرده و اجازه می‌دهد تا گیرنده با صفر کردن این خط، تصدیق را ارسال کند. در صورتی که گیرنده به هر علتی (مشغول بودن، قطع بودن اتصال، از

کار افتادن قطعه و ...) نتواند خط SDA را زمین کند، خط به وسیله‌ی مقاومت Pull-up یک شده و به طور پسیو، سیگنال عدم تصدیق ارسال شود. یک از کاربردهای عدم تصدیق، استفاده از آن برای پایان دادن به ارسال داده است. بدین ترتیب که فرستنده برای آخرین بایت دریافتی از گیرنده، تصدیق ارسال نکرده و بدین ترتیب فرستنده پایه SDA را رها کرده و Master وضعیت پایان ایجاد می‌کند.

آشنایی با بسته‌های (Packet) آدرس و داده

در باس I²C، هر قاب شامل تعدادی بسته است که معمولاً یکی از آنها بسته‌ی آدرس و بقیه بسته‌ی داده می‌باشند. همانطور که در شکل زیر دیده می‌شود، هر قاب اطلاعاتی با یک شرایط آغاز و پایان محصور شده است که از این پس به طور خلاصه شرایط شروع را S، شرایط پایان را P و بازشروع را Sr می‌نامیم.

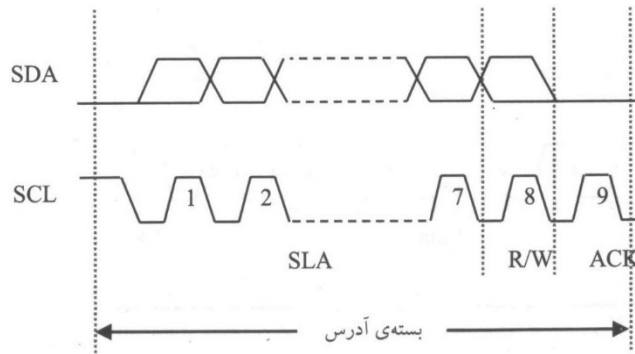


برای برقراری یک جلسه‌ی بین Master و Slave ابتدا باید توسط Master یک شرایط شروع ایجاد شود. پس از این رویداد، Master کنترل باس را بر عهده می‌گیرد و باید ضمن تولید پالس کلاک، Slave مورد نظر را نیز آدرس‌دهی کند. آدرس‌دهی از طریق بسته‌ی آدرس انجام می‌شود و در صورت تصدیق Slave، امکان برقراری ارتباط با آن فراهم می‌شود.

برای روشن شدن موضوع، به بررسی محتوی بسته‌های آدرس و داده می‌پردازیم.

• بسته‌ی آدرس

در بسته آدرس، ابتدا MSB ارسال شده و همانطور که در شکل زیر می‌بینید این بسته به طول ۹ بیت است که شامل هفت بیت آدرس، یک بیت Read/Write و یک بیت تصدیق است.



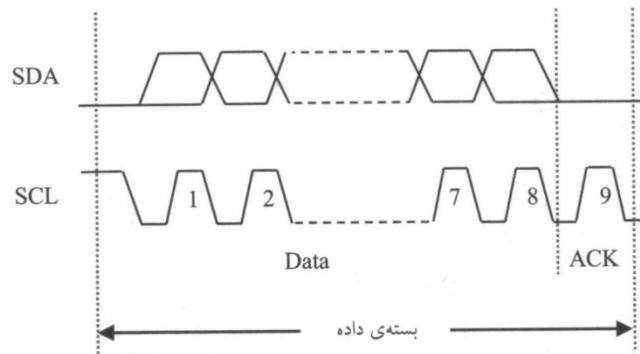
همانطور که اشاره شد در نسخه‌های مختلف I2C از آدرس ۷ و ۱۰ بیتی پشتیبانی می‌شود اما از آنجایی که در میکروکنترلر AVR همواره طول آدرس هفت بیت، بسته‌ی معرفی شده در این بخش نیز شامل ۷ بیت است. با توجه به ۷ بیتی بودن آدرس به نظر می‌رسد که امکان تخصیص حداکثر ۱۲۸ Slave Address (SLA) به وسایل متصل به باس وجود داشته باشد اما دو محدودیت زیر موجب کاهش این تعداد می‌شود.

۱- آدرس‌هایی به فرم 1111 XXXX (مانند 1111 0000، 1111 1010 و ...) برای کاربردهای آتی ذخیره شده‌اند و تنها ۱۲۰ آدرس در اختیار است. همچنین آدرس 0000 0000 مربوط به فراخوانی عمومی (General Call) است که تمام Slave‌ها می‌توانند به آن پاسخ دهند.

۲- حداکثر ظرفیت خازنی مجاز بر روی باس I2C، ۴۰۰ پیکوفاراد است. همان طور که می‌دانید یک IC دارای ظرفیت خازنی معادلی روی هر پایه است، با قرار گرفتن وسایل مختلف بر روی باس، این خازن‌ها موازی شده و ظرفیت معادل افزایش می‌یابد. زمانی که ظرفیت به ۴۰۰ پیکو فاراد برسد (علیرغم وجود فاصله تا رسیدن به ۱۲۰ آدرس مجاز) امکان افزودن وسیله‌ی جدید به خط وجود ندارد.

• بسته‌ی داده

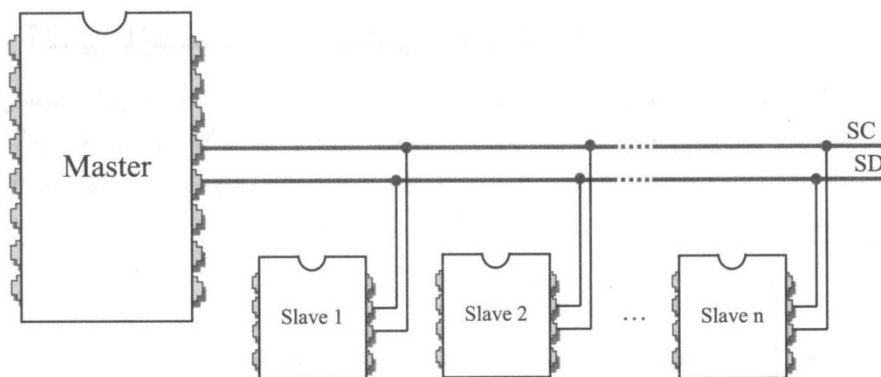
در بسته داده نیز ابتدا MSB ارسال شده و همانطور که در شکل زیر مشخص است، این بسته شامل ۹ بیت می‌باشد. I2C یک استاندارد ۸ بیتی است و بنابراین داده به صورت بایت به بایت منتقل می‌شود، پس از ارسال هر بایت لازم است تا گیرنده در سیکل نهم کلاک با صفر کردن خط SDA برای فرستنده تصدیق ارسال کند.



زمانی که گیرنده، آخرین بایت را دریافت می‌کند و یا به دلایلی نمی‌تواند داده‌ی جدیدی را بپذیرد، لازم است تا با عدم ارسال تصدیق (NACK) و رها کردن خط SDA در سیکل کلاک نهم، اتمام ارتباط را به فرستنده اعلام کند.

آشنایی با روش انتقال داده در باس I²C

همانطور که در شکل زیر مشاهده می‌کنید، باس I²C شامل یک (و یا چند) Master و تعدادی وسیله‌ی Slave است. هر کدام از Slave‌ها دارای یک آدرس منحصر به فرد می‌باشند که قبلاً توسط طراح سیستم مشخص شده است. به منظور شروع ارتباط، Master یک شرایط شروع بر روی باس ایجاد کرده و همزمان با هفت پالس کلاک بعدی، آدرس مربوط به Slave مورد نظر (SLA) را ارسال می‌کند. در پالس کلاک هشتم، با ارسال صفر یا یک توسط Master، جهت ارتباط (R/W) مشخص می‌شود. در صورتی که همزمان با این کلاک، خط داده صفر باشد، Master قصد ارسال داده به Slave را دارد و در صورتی که یک باشد، Master داده را از Slave دریافت می‌کند. این جهت تعیین شده، تا انتهای ارتباط (ایجاد شرایط پایان یا بازشروع) ثابت بوده و امکان تغییر آن وجود ندارد.

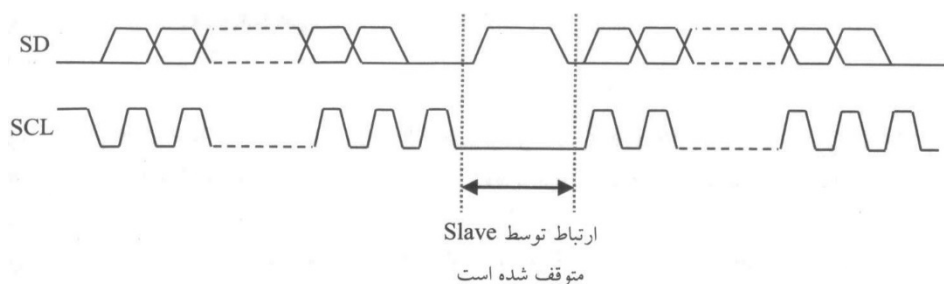


در نهمین پالس کلاک، Slave آدرس دهی شده وظیفه دارد تا با صفر کردن خط داده به Master پاسخ دهد (ACK). در واقع، اولین تاثیر Slave بر روی باس در این لحظه است و تا پیش از این، Slave داده‌ای ارسال شده توسط Master را بیت به بیت قرائت می‌کند تا در انتها با آدرس خود مقایسه نماید.

پس از اتمام بسته‌ی آدرس، در صورتی که Master از Slave تصدیق دریافت کرده باشد، با توجه به بیت R/W در یکی از وضعیت‌های Master Transmitter یا Master Receiver قرار گرفته و Slave نیز بسته به بیت R/W در یکی از دو حالت Slave Transmitter یا Slave Receiver خواهد بود. بنابراین فرستنده (که ممکن است Master یا Slave باشد) بایت به بایت داده را برای گیرنده ارسال می‌نماید و در انتهای هر بایت، از آن تصدیق دریافت می‌کند.

عدم دریافت تصدیق (NACK) نشان دهنده‌ی این است که گیرنده نمی‌تواند یا قصد ندارد بایت بعدی را از فرستنده دریافت کند و بنابراین Master می‌تواند با ایجاد شرایط پایان به ارتباط پایان دهد و یا وضعیت بازشروع ایجاد نماید.

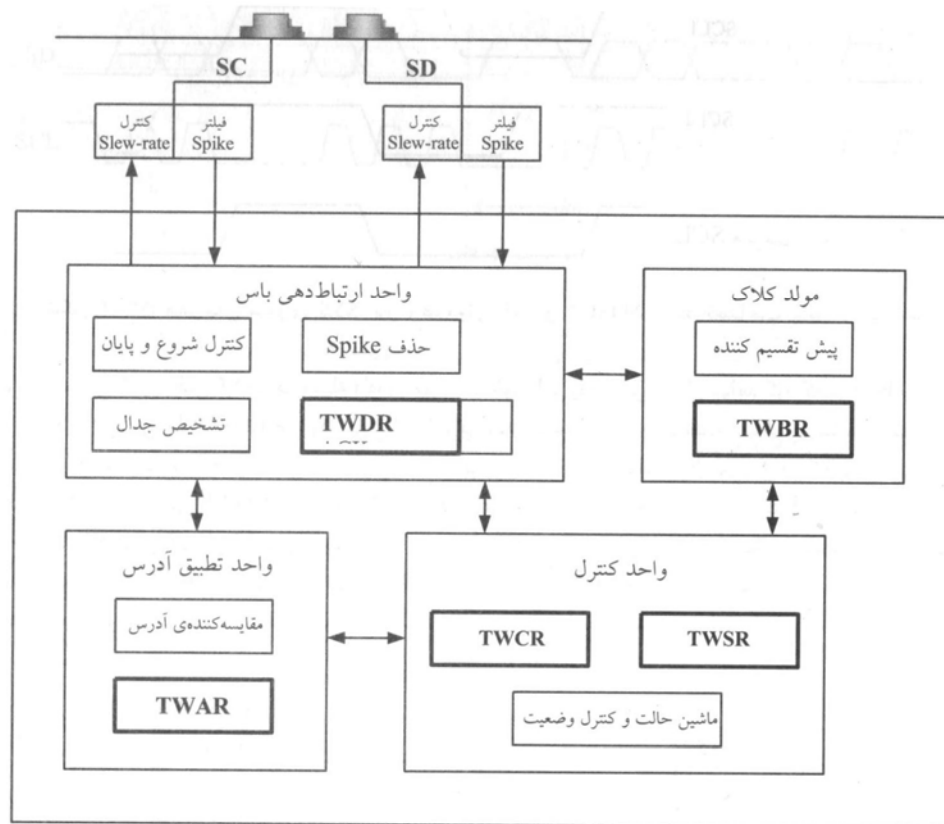
یکی از ویژگی‌های جالب باس I²C امکان ایجاد دست‌تکانی (Handshake) بوسیله سیگنال کلاک است. در صورتی که سرعت وسیله‌ی Slave کمتر از Master باشد و یا در فواصل بین دریافت یا ارسال هر بایت، نیاز به اتمام یک سری عملیات باشد، Slave می‌تواند با صفر کردن خط SDA، Master را در حالت انتظار قرار دهد. این ویژگی که از Wired-AND بودن باس I²C حاصل می‌شود این امکان را می‌دهد تا در مدتی که Slave مشغول است با صفر کردن خط SCL مانع از ایجاد کلاک بوسیله‌ی Master شود (شکل زیر)



آشنایی با واحد TWI در میکروکنترلر AVR

برای ارتباط‌دهی یک میکروکنترلر با باس I²C می‌توان بوسیله نرم‌افزار پروتکل را پیاده‌سازی کرد و یا یک کنترلر I²C (مانند مدار مجتمع PCF8584) استفاده کرد. در روش اول بخش عمده‌ای از وقت پردازنده اشغال

شده و سرعت ارتباط نیز معمولاً پایین است و در روش دوم اگر چه این مشکلات وجود ندارد اما نیاز به یک سخت‌افزار خارجی و ارتباطی آن با میکروکنترلر است.



شکل ۱۳.۱۶: واحد TWI در قطعه ی ATmega32

خوشبختانه در میکروکنترلرهای AVR، کنترلر I^2C به صورت داخلی تعبیه شده است و در نتیجه معمولاً نیاز به استفاده از دو روش ذکر شده نمی‌باشد.

از آنجایی که شرکت اتمل (که سازنده میکروکنترلرهای AVR است)، I^2C را واسطه دوسیمه یا TWI به منظور مطابقت با مستندات میکروکنترلر می‌نامد به منظور مطابقت با مستندات میکروکنترلر AVR در ادامه واحد TWI یا Two Wire Interface را به همین نام خطاب کرده و در سایر موارد از لفظ I^2C استفاده خواهیم کرد.

در شکل فوق واحد TWI میکروکنترلر ATmega32 نشان داده شده است. همانطور که ملاحظه می‌کنید این سخت‌افزار شامل واحد ارتباط دهی باس، مولد نرخ کلاک، واحد تطبیق آدرس، واحد کنترل بوده، پایه‌های SCL و SDA است که به بررسی هر یک از این موارد می‌پردازیم.

• پایه‌های SDA و SCL

این پایه‌ها ارتباط بین واحد TWI و باس I²C را فراهم می‌کنند. SDA و SCL به ترتیب عملکرد دوم PC1 و PC0 بوده و هر یک دارای یک کنترل کننده‌ی slew-rate و فیلتر Spike می‌باشند. کنترل کننده‌ی slew-rate که نرخ تغییرات دامنه‌ی پالس‌های ایجاد شده را محدود می‌کنند در راه‌انداز خروجی قرار گرفته و موجب بهبود مشخصات الکتریکی سیگنال ارسال شده، می‌شود. فیلتر Spike نیز در طبقه ورودی بوده و با زدودن پالس‌های سوزنی با عرض کمتر از ۵۰ نانو ثانیه به کاهش حساسیت نسبت به نویز کمک می‌کند.

همانطور که اشاره شد، باس I²C برای عملکرد صحیح نیاز به دو مقاومت Pull-up دارد. در برخی موارد می‌توان با فعال کردن مقاومت Pull-up پایه‌های PC1 و PC0 نیاز به مقاومت Pull-up خارجی را مرتفع کرده. در میکروکنترلر ATmega32 مقدار مقاومت Pull-up داخلی بین ۲۰ تا ۵۰ کیلو اهم بوده و بنابراین می‌توان از آن در فرکانس‌های پایین استفاده کرد.

• واحد ارتباط دهی باس

این واحد شامل شیفت رجسترهای داده و آدرس، کنترلر شروع و پایان و تشخیص جدال بین Masterها است. رجیستر TWDR شامل آدرس یا داده‌ای است که باید ارسال شود و یا دریافت شده است. علاوه بر رجیستر هشت بیتی TWDR، این واحد شامل یک رجیستر تک بیتی مربوط به تصدیق است که به طور مستقیم در دسترس نبوده و ACK یا NACK را نگهداری می‌کند.

واحد ارتباط دهی باس شامل سخت‌افزار تشخیص جدال نیز می‌باشد. این واحد به طور پیوسته مقدار ارسال شده توسط واحد TWI را با مقدار ایجاد شده روی باس مقایسه کرده و در صورت بروز جدال آن را تشخیص می‌دهد. در صورتی که Master طی حکمیت، مغلوب شده و کنترل باس را از دست بدهد این سخت‌افزار، واحد کنترل را در جریان قرار می‌دهد.

• مولد نرخ کلاک

این واحد، در وضعیت‌های Master Transmitter و Master Receiver فرکانس پالس ایجاد شده بر روی SCL را کنترل می‌کند. واحد TWI دارای پیش تقسیم کننده‌ای است که تقسیمی از فرکانس کلاک سیستم را برای مولد کلاک فراهم می‌کند.

• واحد تطبیق آدرس

این واحد مقدار آدرس ارسال شده بر روی خط را با آدرس موجود در رجیستر TWAR مقایسه کرده و در صورت برابری، واحد کنترل را از تطبیق آدرس آگاه می‌کند تا عملکرد مناسب انجام شود.

• واحد کنترل

این واحد یکی از پیچیده‌ترین اجزاء واحد TWI است که پیوسته باس I2C را مونیتور کرده و پاسخ مناسب را ایجاد می‌کند. با روی دادن موارد زیر بر روی باس فلگ TWI یک شده و واحد کنترل، عملکردی متناسب با آن رویداد خواهد داشت:

- پس از ارسال وضعیت شروع یا بازشروع
- پس از ارسال بسته‌ی آدرس
- پس از مغلوب شدن در جدال با Master دیگر
- پس از آدرس‌دهی شدن بوسیله‌ی یک Master دیگر و یا دریافت آدرس فراخوانی عمومی
- پس از دریافت یک بسته‌ی داده
- پس از ارسال یک بسته‌ی داده
- پس از دریافت شرایط پایان
- زمانی که به علت ایجاد شرایط آغاز یا پایان غیرمجاز خطایی باس روی می‌دهد.

آزمایشی که باید انجام دهید:

سخت‌افزاری با استفاده از میکروکنترلر ATmega32 و ساعت DS1307 طراحی گردیده است. در محیط کدویشن، I2C را فعال نمائید و در روی tab ایجاد شده DS1307 را انتخاب کنید و آنرا در حالت انتخاب قرار دهید (لازم است که I2C Port را روی PORTC قرار دهید). سپس با استفاده از توابع `rtc_get_date()` و `rtc_get_time()` زمان و تاریخ را خوانده و روی صفحه LCD، 4x20 در خطوط اول و دوم نمایش دهید (همانند آنچه در برنامه نمونه مشاهده مینمایید).