

سوال اول

(الف)

- Full-Duplex
این مفهوم در پروتکل SPI استفاده شده است و به آن معناست که میتوان به صورت همزمان هم دیتا فرستاد و هم دریافت کرد (Send & Receive) مانند یک تلفن. این برخلاف Half-Duplex است که فقط یکی از این دو کار در واحد زمان انجام میشود.
- Slave Receive
این مفهوم زمانی است که Master صرفاً دیتا بفرستد و Slave صرفاً دیتا دریافت کند و نه برعکس. این پروتکل برای I2C است. (Writer/Slave Receiver Master)

(ب) مفهوم Open Drain یا به عبارتی Two Bidirectional Open Collectors مانند یک سویچ عمل میکند که میتواند به Ground یا Vcc متصل باشد و مقدار آن به ترتیب 0 یا 1 شود. این روش در پروتکل I2C استفاده شده است به طوری که هر دو سیم SDA و SCL به منبع Vcc متصل بوده و همواره مقدار آنها برابر 1 است تا زمانی که یکی از Device ها آن را تغییر دهد.

سوال دوم

(الف) با خواندن بیت مورد نظر از PIO_ISR میتوان متوجه شد کدام پین وقفه را درخواست داده است. (رجیستر PIO_ISR مانند آرایه ای از بیت ها است، شماره بیت مورد نظر اطلاعات مربوط به پین را ذخیره کرده است)

(ب)

- PIO_ODSR = 1
- PIO_MDSR = 0
- PIO_OSR = 0

سوال سوم

(الف) در PIO Controller قطعه ای به نام PMC یا Power Management Controller ورودی کلاک را بر دستگاه کنترل میکند (میتواند آن را قطع یا وصل کند). نوشتن در رجیستر های آل نیازی به پالس کلاک ندارد پس برای صرفه جویی در مصرف میتوان آن را غیر فعال کرد. اما غیر فعال بودن کلاک باعث میشود همه ی بخش های PIO قابل استفاده نباشد مانند خواندن از پین ها، Interrupt modes، Input change Interrupt و ... (این سه قطعه نیاز به کلاک دارند)

(ب)

- نوشتن بر روی رجیستر های PIO Controller: فعال
- خواندن سطح منطقی پین های خروجی: غیر فعال
- استفاده از پین ها برای تولید وقفه: غیر فعال
- Pull-up کردن پین ها: فعال

سوال چهارم

مهم ترین تفاوت این دو روش موازی بودن عملکرد Slave ها است به طوری که در روش دوم Slave ها به گونه ای موازی شده اند یعنی فعال بودن یکی، به فعال بودن دیگری تاثیر نمی گذارد اما در روش اول Slave ها به صورت سری به هم متصل شده اند به گونه ای که اگر Master اولین Slave را فعال/غیر فعال کند بقیه Slave ها هم فعال/غیر فعال میشوند. (زیرا خروجی SD یکی به SD دیگری متصل است). پس در روش اول یا همه Slave ها فعال اند و یا غیر فعال، در صورتی که در روش دوم میتوان یک Slave خاص را فقط فعال کرد.

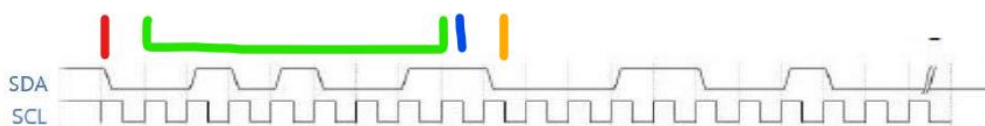
سوال پنجم

(الف)

چه داده ای؟ 01

با کدام آدرس؟ 0101001

به صورت خواندن یا نوشتن؟ 1 (خواندن)



ب) در پروتکل I2C در ابتدا یک بیت Start، 7 بیت آدرس و یک بیت خواندن/نوشتن داریم. بعد از آن هر بابت و بیت ACK آن تکرار میشود. پس 9 بیت برای شروع کار داریم و 8 بیت 8 بیت (یکی ACK و بقیه دیتا) را داریم. در نتیجه برای I2C در کل داریم:

$$9 + (115 * (7 + 1)) = 929$$

ولی در SPI تنها بیت های دیتا منتقل میشوند و نه چیز بیشتری پس بازدهی آن 1 است. بازدهی SPI از I2C بیشتر است.

سوال ششم

الف) از مشکلات این پدیده تغییرات بسیار سریع و ناگهانی در وضعیت دستگاهی است که این پدیده برایش رخ میدهد. برای مثال در چراغ میتواند برای مدت کوتاهی به سرعت خاموش و روشن شود. این پدیده آسیب بسیاری به دستگاه ها وارد میکند. برای جلوگیری از آن میتوان برای مدت زمان خیلی کوتاهی خروجی را نادیده گرفت. به عبارتی برای مدت کوتاهی صبر میکنیم تا نوسانات سیگنال Stable بشوند.

ب) این مفهوم در پروتکل I2C مورد استفاده قرار میگیرد و در آن Slave به Master اطلاع میدهد که صبر کند (wait) زیرا Slave به زمان بیشتری برای پردازش نیاز دارد (این پردازش میتواند برای دیتا ورودی باشد یا پیش پردازش های دیتایی که قرار است فرستاده شود) به عبارتی Slave نمیتواند با سرعت کلاک Master همراه باشد (Keep-up).

برای این عمل بعد از بیت ACK و قبل از اولین بیت بعدی باید Slave یک بیت به Master بفرستد تا اطلاع رسانی کند.
(این اطلاع رسانی بین بیت های 2 تا 9 نباید اجرا شود)