

SPI

Seriālais perifērijas interfeiss (SPI) ir vienkāršs no punkta uz punktu orientēts interfeiss kas ir bāzēts uz vedēja – sekotāja principa. Tas sniedz pilnu duplexu komunikāciju starp galveno un vienu vai vairākiem padotajiem. Interfeiss sastāv no četrām līnijām:

MOSI (galvenais ārā padeotais iekšā - master out slave in) : šo līniju izmanto lai pārnestu datus no galvenā uz padoto.

MISO (galvenais iekšā, padotais ārā – master in, slave out) : šo līniju izmanto lai pārnestu datus no padotā uz galveno.

SCK (sistēmas pulkstenis – system clock) : šī līnija tiek izmantota lai izmantojot galveno tiktu pārraidīts sistēmas pulksteņa signāls.

SS (padotā izvele – slave select) : šo līniju izmanto lai izvēlētos padoto.

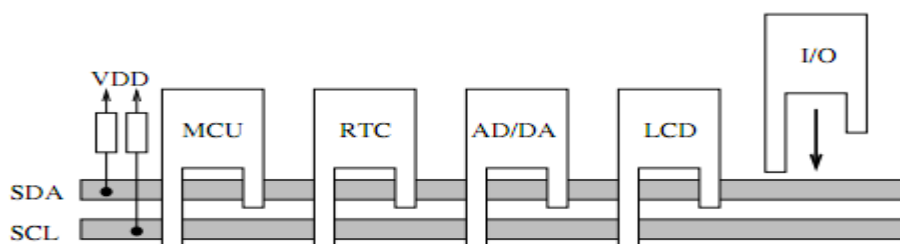
Attēlā 3.6. ir parādīti pamata principi SPI interfeisam. Abi, gan galvenais, gan padotais ir ar iekšējās pārbīdes reģistriem kas tiek operēti izmantojot SCK. Pie katra pulksteņa signāla msb (vai lsb, tas ir rediģējams) galvenā SPI reģistrs tiek pārņemts ārā uz MOSI līniju un pārņemts uz padoto SPI reģistru kā lsb. Tajā pat laikā, padotā msb tiek pārraidīts izmantojot MISO līniju uz galvenā reģistra lsb. Pēc 8 šādiem cikliem galvenais un padotais ir apmainījušies ar visiem 8 bitiem savos reģistros.

Galvenajam ir jāpiešķir noteikta adrese padotajam nomainot SS uz zemu. Tātad mēs varam savienot divus padotos pie viena SPI ja mēs ļaujam vienam no tiem reaģēt tieši ar SS, no kā izriet ka otrs pirmais noliedz līniju. Uzstādot SS uz zemu līniju atbild augsti, galvenais tad var izvēlēties pirmo atbildi kā otro padoto.

Ja galvenais ir sagatavots lai izmantotu vairāk I/O savienojumus kā tam ir, tad padoto numurs var tik palielināts līdz 2^n kur n adresējas ar ārējā dekodera palīdzību.

3.3 IIC (I²C)

Inter- IC kopne (IIC) ir sinhrona kopne, kas darbojas ar vadītāja- izpildītāja funkciju. Tas lieto divas stieples – SCL (Sērijveida Laika Līnija) un SDA (Sērijveida Datu Līnija) pus duplexai komunikācijai. Protokolu izstrādāja Philips, un tas tiek plaši izmantots (īsās distances) komunikācijai starp vienu vai vairākiem kontrolieriem un periferiālajām ierīcēm. Protokols iekļauj kopnes arbitrāžas mehānismus, un tas ļauj kopēji pastāvēt dažādiem vadītājiem. Parasti vadītāja loma gulstas uz mikrokontrolieriem visās periferiālajās ierīcēs kā parasti izpildītāji. Sistēmās ar dažādiem mikrokontrolieriem jūs varat izvēlēties katram kontrolierim, vai, lai tas ir vadītājs vai izpildītājs. Vienīgais nosacījums ir, ka ir jābūt vismaz vienam vadītājam sistēmā. Viens no galvenajiem ieguvumiem IIC kopnei ir labā elastība. Jaunās ierīces var pievienot kopnei tikai ar savienošanu, sk. 1 attēlu. Nav noteiktu ierobežojumu ierīču skaitam, kas savienots ar kopni tik ilgi, kamēr netiek pārsniegts 400pF kapacitātes maksimums.



Zīmējums 3.7: IIC kopnes pamatkonfigurācija

IIC kopne atbalsta gan 7-bitu, gan 10-bitu adreses. 7-bitu adresēšanā visas ierīces uz kopnes tiek identificētas ar 7-bitu adresi, kuras daļa (t.i. augstākais 4 biti) ir stingri kodēta no ražotāja puses. Atlikušos bitus var programmēt galvenie dizaineri. Adreses $(0000X\ X\ X)_2$ un $(1111X\ X\ X)_2$ tiek rezervētas, atstājot 112 potenciālās ierīču adreses. Lai palielinātu ierīču adrešu skaitu, protokols vēlāk tika paplašināts uz 10-bitu adresēm. Rūpējās par to, lai atliktu savienošana ar oriģinālo 7-bitu adresēšanas režīmu, tādējādi 7-bitu adrešu ierīces var tikt lietotas jaunā adresēšanas shēmā. Šajā sadaļā mēs fokusēsimies uz 7-bitu adresēšanas shēmu.

Mēs neesam līdz šim skaidri minējuši, bet tas ir acīmredzams, ka periferiālajām ierīcēm ir sevī jāiekļauj IIC (izpildītājs) interfeiss, lai izveidotu iespējamu „iespraud- un – spēlē” īpašību 3.7 attēlā. Ražotāji, kas vēlas aprīkot savas ierīces ar IIC kopnes interfeisu, jāiegādājas licence no Philips, un tad jāsavieno protokols savās ierīcēs. Līdzīgi, mikrokontrolieriem dažreiz jau ir integrētais IIC modulis, kas izpilda protokolus. Kā arī citi interfeisa moduļi, tas piedāvā programmētājiem augsta līmeņa pieejas funkcijas. Tas arī ļauj programmētājam izvēlēties starp vadītāja un izpildītāja režīmiem, un definēt komunikācijas parametrus, tādus, kā transmisijas ātrums vai saņēmēja adrese.

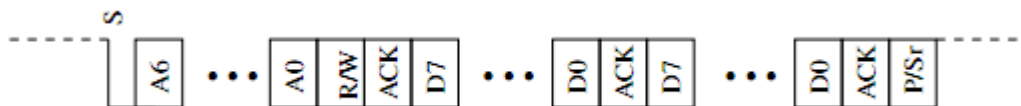
Datu pārraide

IIC ir patstāvīgi nobeidzoša kopne, voltāžas līmeņi tiek noteikti ar respektu pret kopējo pamatu. Zema līmeņa ievades voltāža ir no 0.5 līdz $0.3V_{DD}$ līmenī. Voltižēšana augsta līmeņa voltāžai ir no $0.7V_{DD}$ līdz $V_{DD} + 0.5V$. Zems līmenis datu līnijā attiecas uz loģisko 0, augsts līmenis attiecas uz loģisko 1.

Kopš stieples ir pievienotas ārējai ieejas pretestībai (no 1-10 k Ω rangā), augsts līmenis nekad netiek sasniegts. Tādējādi, sūtītājs vienkārši nostabilizē tā izejas, un liek stieplei iet augšup ārēji. Atzīmējam, ka protokols ir atkarīgs no tā īpašībām, tādējādi, ja vēlaties ieprogrammēt protokolu manuāli (procedūra, kura saukta arī kā bitu trieciens), jums nevajadzētu darbināt augsto līmeni. Ja jums ir jāizliek loģiskais 1, ievadiet kontroliera pin ieejas vietā. Zems līmenis tiek darbināts kā parasti, tāpēc, ierakstiet tikai 0 izejas pin.

Šī asimetrija, kas ir starp augstu un zemu līmeni, veidojas līnijā, kurā ir dominējošs un recesīvs stāvoklis: Ja ierīce izvada 0, un tādējādi darbina līniju lēni, tā paliek lēna, pat, ja kādas citas ierīces izeja ir 1 (t.i. nostāda to izeju). Tādējādi, 0 vienmēr ir pārsvars pār 1 (zemais ir dominējošais, augstais ir recesīvais). Šī īpašība bieži tiek saukta kā stieple – AND (jo visiem sūtītājiem ir izeja 1 līnijai, lai būtu 1) un dažreiz stieple – NOR (iespējams, jo no skatu punkta atvērta kolektora izeja, ja viena ierīce pagriežas uz tās izejas tranzistoru, tad kopne darbojas zemāk). Kā mēs varēsim apskatīt vēlāk, protokols izmanto tā īpašības, lai kontrolētu ātrumu un kopnes arbitražu.

IIC paketes galvenais izklāsts ir attēlots 3.8. attēlā.

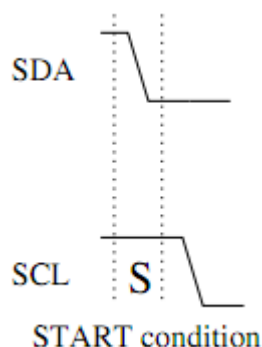


Zīmējums 3.8: IIC freims

To brīvajos stāvokļos gan SCL, gan SDA ir augsts. Transmisiju aizsāk vadītājs, kas izliek laika signālu uz SCL un nosaka starta stāvokli (S) uz SDA. Tad vadītājs uzliek adresi uz kopnes un nosaka, kura no abām ir lasīšanas pieeja (tas ir, adresētajai ierīcei ir jāpārvieto dati) un rakstīšanas pieeja (vadītājs pārvieto datus uz ierīci). Pēc R/W bita, izpildītājs sūta atpazīšanu, lai noskaidrotu, vai tas ir atpazinis tā adresi. Atkarībā no datu virziena, gan vadītājs, gan izpildītājs tagad var pārvietot arbitražas skaitu datu baitu. Katrs baits tiek atpazīts no saņēmēja puses (ar pēdējā baita izņēmumu, ja vadītājs ir saņēmējs). Transmisijas beigās, vadītājs ģenerē STOP stāvokli (P), lai parādītu, ka kopne tagad ir brīva, vai tas var paturēt kopni, sūtot atkārtotu START stāvokli (Sr), kas izbeidz konkrēto transmisiju un tajā pašā laikā uzsāk jaunu.

Starts un Atkārtotais Starts

START stāvoklis (S) ir parādīts 3.9 attēlā. Tas tiek raksturots ar krītošu asumu uz SDA līnijas, kamēr tiek veidoti augsti līmeņi SCL līnijai. Atzīmējam, ka tikai START un STOP stāvokļi maina SDA līmeni, kamēr tiek darbināts SCL stāvoklis. Visas normālas datu transmisijas iekļauj apliecinājuma izmaiņas līmenī, veicot zemo stāvokli SCL.



Zīmējums 3.9: START stāvoklis.

Atkārtotais START stāvoklis (Sr) ir tāds pats kā normālais START stāvoklis. Tas aizvieto STOP/START stāvokļa pāri un tiek sūtīts tā vietā, ja vadītājs nosaka tūlītēju kopnes lietošanu atkal. Patstāvīga vadītāja sistēmā, Sr vienkārši saglabā vienu laika ciklu. Multivadītāju sistēmā Sr novērš arbitrāžas fāzi (sk. Nodaļu 3.3.3), un, tādējādi nodrošina, ka esošais transmisijas vadītājs patur kopni.

Adrese

7 – bitu adrese tiek sūtīta msb vispirms. Kā ar visiem normāliem datu bitiem, SDA līmenis tiek mainīts SCL zemās fāzes laikā, un tiek nolasīts augstās fāzes laikā.

Virziena kontrole

Pēc septiņiem adreses bitiem, vadītājs pabeidz baitu ar astoņiem bitiem (R/W), kas nosaka virzienu sekojošajām transmisijām. Ja R/W ir augsts, tad vadītājs vēlas nolasīt datus no adresētā izpildītāja. Ja bits ir zems, tad vadītājs paredz sūtīt datus izpildītājam.

Atzīmējam, ka katra transmisija kopnē tiek uzsākta no vadītāja puses, kas sūta izpildītāja adreses. Ja R/W ir augsts, izpildītājs sūta tā apstiprinājumu. Pēc tam, datu virziens mainās un izpildītājs uzsāk datu transmisiju.

Apstiprināšana

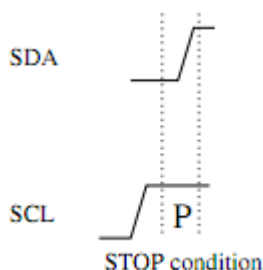
Pēc katriem 8 bitiem saņēmējs sūta apstiprinājumu (ACK), lai noteiktu, vai tas ir saņēmis datus. ACK tiek sasniegts, uzstādot zemu SDA. Vienīgais izņēmums ir pēdējais apstiprinājums no vadītāja puses, ja tas ir saņēmējs: Šajā gadījumā, vadītājs neapstiprina pēdējo baitu, tas ir, SDA paliek augsts. Augsts nozīmē pārsūtīšanas izpildītājam, ka transmisijas beigas ir sasniegtas. Tad izpildītājs palaiž datu līniju, lai ļautu vadītājam nodot atkārtoto startu vai apstādināt nosacījumu.

Dati

Datu biti tiek nodoti tāpat kā citi biti, un katru baitu ir jāapstiprina no saņēmēja puses. Dati vispirms ir nodoti msb. Nav ierobežots datu baitu skaits, kuri var tikt nodoti vienā reizē.

Stop

3.10 attēls parāda STOP stāvokli (P). Tas attēlo START stāvokli, tādējādi SDA tagad pāriet no zema uz augstu, kamēr SCL ir augsts. STOP stāvoklis (P) tiek nodots no vadītāja, ja tas vēlas nodot kopni. Cik ātri vien STOP stāvoklis tiek nosūtīts, tik ātri kopne ir tukša, un var tikt pieprasīta no jeb kura vadītāja, sk. nodaļu 3.3.



Zīmējums 3.10: STOP stāvoklis (P)

Ātruma kontrole caur izpildītāju

Kopš nav veida kā pārvarēt transmisijas ātrumu starp vadītāju un izpildītāju, vadītājs vienkārši nodod datus ar fiksētu ātrumu. Protams, ātrums tiek pielāgots izpildītāja ierīču specifikācijām, bet vēl jo projām var notikt tas, ka izpildītājam vajag mazliet laika, lai apstrādātu datus, kas tiek sūtīti no vadītāja. Tādējādi, mums vajag nozīmes, lai ziņot vadītājam, lai tas mazliet pagaida. ICC protokols diezgan gudri izmanto laika līnijas dominējošo/recesīvo dabu, lai sasniegtu ātruma kontroli ar izpildītāju. Ļaujiet mums vispirms definēt laika signāla divas fāzes: zemā fāze ir periods, kas darbojas, kad laika signāls ir zems. Līdzīgi, augstā fāze ir periods, kas notiek, kad laika signāls ir augsts. Lai ģenerētu signālu, vadītājam jāuzsāk fāzi ar piemērotu signāla asumu (augošs vai krītošs), un tad nogaidīt, kad pienāk laiks ģenerēt nākamo asumu, tāpat kā PWM signāli ir ģenerēti ar taimeri. Tagad, lai iegūtu izpildītāja laika kontroli, vadītājs neģenerē tā laika signālu uz aklo, bet izlasa atpakaļ SCL un uzsāk tikai laika noteikšanu zemā vai augstā fāzē pēc tam, kad ir noteikta krītošs vai augošs asums uz SCL. Kā rezultāts, jeb kad, kad izpildītājam vajag vairāk laika, tas vienkārši var noteikt zemu laika līniju. Kopš zemais ir dominējošais stāvoklis, tas pagarinās konkrēto zemo laiku un tādējādi kavēs vadītāju, kamēr izpildītājs realizēs SCL. Kamēr vadītājs gaida, kamēr būs pieaugošs asums pirms laika noteikšanas augstās fāzes, visi atbilstošie laika cikli tiek kavēti, bet ne novērsti.

Šis mehānisms rada to, ka IIC ierīces ir diezgan elastīgas tikmēr, kamēr bitu temps tiek noteikts. Tādējādi trīs ātruma režīmi rada maksimālos reitingus, bet aktuālais transmisijas ātrums kopnē var būt patvaļīgi zems tikmēr, kamēr IIC kopnes laika noteikšanas ierobežojumi (kas nenosaka laika ciklu darbību, bet pārsvarā dalās ar atkarību starp SDA un SCL līnijām) tiek sasniegti.

Multivadītāja Režīms

Lai darbotos multivadītāju sistēmas, mums vajag dažas nozīmes, lai kontrolētu kopnes pieeju, un, lai sinhronizētu dažādos laika signālus. IIC kopnes specifikācija apstrādā stieples – UN īpašības SDA un SCL līnijās, lai sasniegtu šos mērķus.

Apskatīsim vispirms SCL: pieņemot, ka sākumā katrs vadītājs ģenerē tā laika signālu neatkarīgi no citiem vadītājiem, un novieto to uz SCL. Balstoties uz stieples – AND īpašībām, pirmais vadītājs, lai ģenerētu krītošu asumu, nosaka SCL uz zemu. Kopš vadītājs nolasa atpakaļ esošo SCL apjomu, kā aprakstīts 3.3.2 nodaļā, un uzsāk laika noteikšanu pie augstas vai zemas fāzes, tikai pēc esošā asuma, kas tiek noteikt uz SCL, pirmais krītošais asums uz SCL izraisa zemas fāzes laika noteikšanu visiem vadītājiem. Tagad viens pēc otra vadītāji pabeigs to zemās fāzes un mēģina uzstādīt SCL kā augstu. Lai vai kā, kamēr kaut viens vadītājs atliek zemu, SCL līnija paliek zema stieples – AND īpašībās. Tādējādi, vadītājs ar garāko zemo fāzi ģenerē zemo fāzi rezultāta SCL signālam. Kad šis vadītājs paliek augsts, SCL signāls pieaug uz augstu un visi vadītāji uzsāk laika noteikšanu to augstajās fāzēs. Pirmais vadītājs, lai pabeigtu tā augsto fāzi, noteiks SCL atkal zemu, efektīvi pabeidzot SCL augsto fāzi. Tādējādi, vadītājs ar īsāko augsto fāzi ģenerē augsto fāzi rezultāta SCL signālu.

Kopnes arbitražā lieto stieples – AND īpašības SDA līnijai. Kad kopne ir brīva, jeb kurš vadītājs var ģenerēt starta stāvokli un uzsākt nodot izpildītāja adresi. Līdzīgi kā ar SCL līniju, vadītājs nolasa atpakaļ SDA līniju, lai pārbaudītu, vai bits, kas bijis uzrakstīts, patiesība ir veidots tā SDA. Atkal, zems līmenis būs dominējošs, tādējādi, vadītāji, kuri rakstīs 1 un lasīs atpakaļ 0, sapratīs, ka citi vadītāji lieto kopni tāpat un pārtrauks nodošanu. Nobeigumā, atliks tikai viens vadītājs (ņemot, ja diviem vadītājiem ir nosūtīta tā pati ziņa, un šajā gadījumā atpakaļ atslēgšana nav nepieciešama).

Paplašinātās adreses

Kā jau iepriekš minējām, oriģinālā 7-bitu adresēšanas shēma tika vēlāk paplašināta līdz 10-bitu adresēm. Lai veidotu saderību ar oriģinālo 7-bitu adresēšanu, viena no rezervētajām 7-bitu adresēm $(11110X X)_2$ tika lietotas, lai izveidotu 10-bitu adresēšanu: Lai uzrakstītu izpildītājam, vadītājs sūta $(11110X X)_2$ adreses lauku un uzstāda R/W uz 0. XX laukā ir divi visnozīmīgākie biti 10-bitu izpildītāja adresē. Katrs izpildītājs pielīdzina šos divus bitus msb savā pašā adresē un apstiprina, vai ir sakritība. Vadītājs tagad sūta atlikušos 8 bitus adresē nākamajā baitā. Tikai viens no iepriekš adresētajiem izpildītājiem atradīs sakritību un apstiprinās. Pēc tam, vadītājs nodos datus izpildītājam.

Lai nolasītu no izpildītāja, vadītājs vispirms sūta $(11110X X)_2$ un uzstāda R/W uz 0. tad tas sūta zemu adresē baitus un atkal gaida apstiprinājumu. Pēc apstiprināšanas, vadītājs ģenerē atkārtoto starta stāvokli un atkal sūta $(11110X X)_2$, bet šoreiz R/W uzstāda uz 1. izpildītājs, kuram pēdējam tas tika adresēts, reaģēs divu msb saskaņai un uzsāks nodošanu nākamajā baitā.