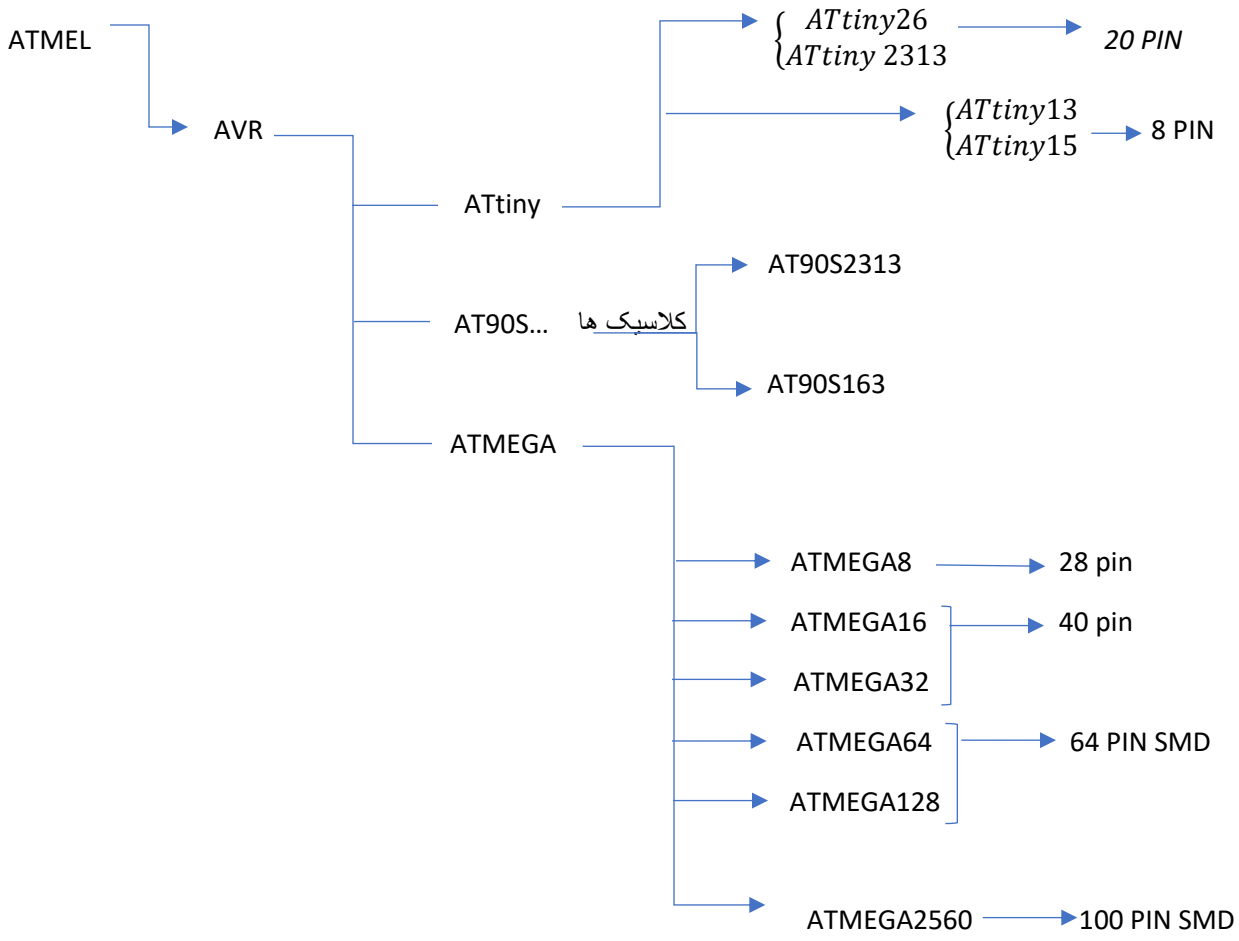
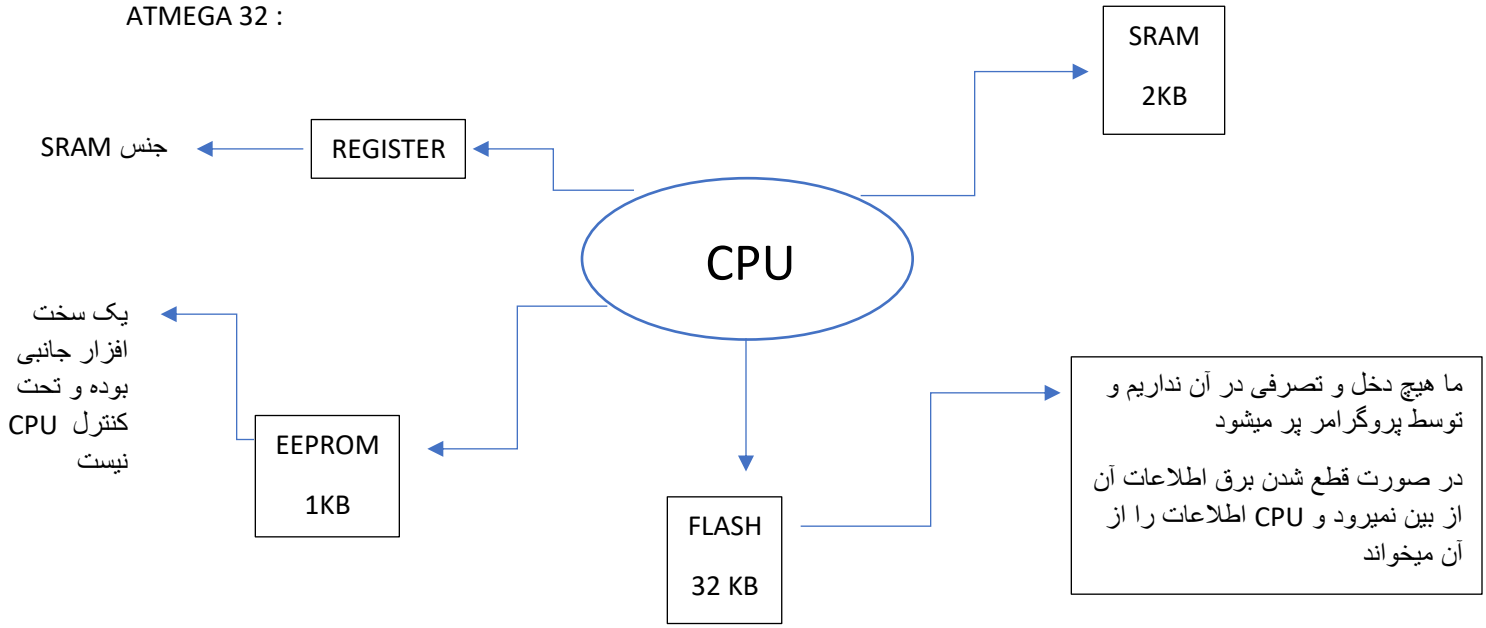

ATMEL
AVR
ATtiny
$\{ATtiny26, ATtiny\ 2313$
20 PIN
$\{ATtiny13, ATtiny15$
8 PIN
AT90S...
کلاسیک ها
AT90S2313
AT90S163
ATMEGA
ATMEGA8
28 pin
ATMEGA16
ATMEGA32
40 pin
ATMEGA64
ATMEGA128
64 PIN SMD
ATMEGA2560
100 PIN SMD

ATMEGA 32 :


SRAM
2KB
REGISTER
جنس SRAM
CPU
یک سخت افزار جانبی بوده و تحت کنترل CPU نیست
EEPROM
1KB
FLASH
32 KB
ما هیچ دخل و تصرفی در آن نداریم و توسط پروگرامر پر میشود
در صورت قطع شدن برق اطلاعات آن از بین نمیرود و CPU اطلاعات را از آن میخواند

انواع مدل های AVR :

پسوند	محدوده تغذیه	محدوده فرکانس
ATMEGA 32	5.5_4.5 V	16MHz
ATMEGA 32A	3.3_5.5 V	16MHz
ATMEGA32 L مناسب دستگاه های پرتابل که باتری متصل میشود	2.7 _ 5.5 V	8MHz
ATMEGA32 V این سری در خانواده های ATtiny وجود دارد	1.8 _ 5.5 V	4MHz

معماری پردازنده های این میکروکنترلرها بر اساس الگوریتم ریسک (RISC) است

RISC : REDUCE INSTRUCTION SET COMPUTER

مدل معماری کاهش یافته دستورات کامپیوتری

این الگوریتم به صورتی طراحی شده تا 90 درصد دستورات ماشین در یک کلاک انجام شود

بدین معنا که در یک ثانیه 16 میلیون عمل انجام میشود. که واحد توانایی سنجی آن MIPS است.

توانایی میکرو ATMEGA32 برابر 16 MIPS است

MIPS : MILION INSTRUCTION PER SECONED

هرچه سخت افزار ما جریان کشی بالایی داشته باشد و در فرکانس بالاتری کار کند نویز پذیری آن کاهش میابد.

در سری L این تراشه ما چند مد خواب داریم :

1 مد Active : 1.1mA

2- مد IDLE : 0.35 mA

3- Power Down < 1uA

مد idle :

کلاك CPU متوقف مي شود.

ميكرو به SPI, ANALOG CAMPARATOR, ADC TIMER/COUNTER,WATCHDOGو وقفه هاي سيستم اجازه كار مي دهد.

برای بیدار نمودن CPU در این مد می توان از وقفه های خارجی و داخلی مانند سرریز تایمر،وقفه مربوط به USART و...بیدار نمود.

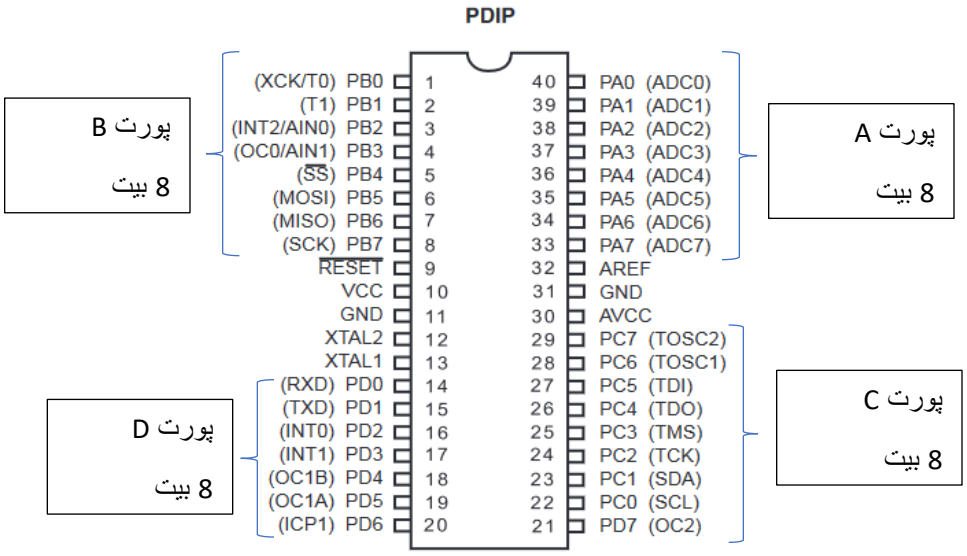
مد power down :

اسيلاتور خارجي متوقف مي شود.

-وقفه هاي خارجي، wire2و WATCHDOG اگر فعال شده باشند به عملياتشان ادامه مي دهند.

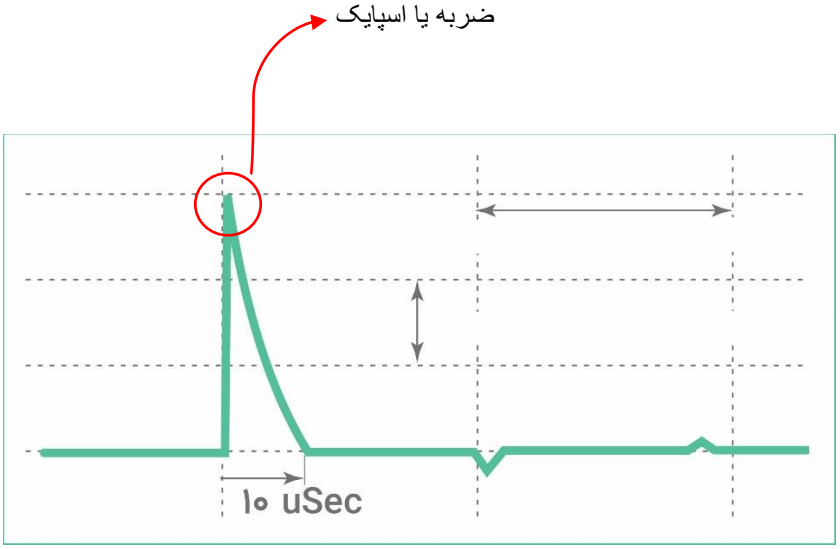
-ريست خارجي ، ريست WATCHDOG ، وقفه خارجي INT2 ،ريست BROWN-OUT ،،وقفه خارجي حساس به سطح INT0 و INT1 مي توانند ميكرو را از اين مد خارج كنند

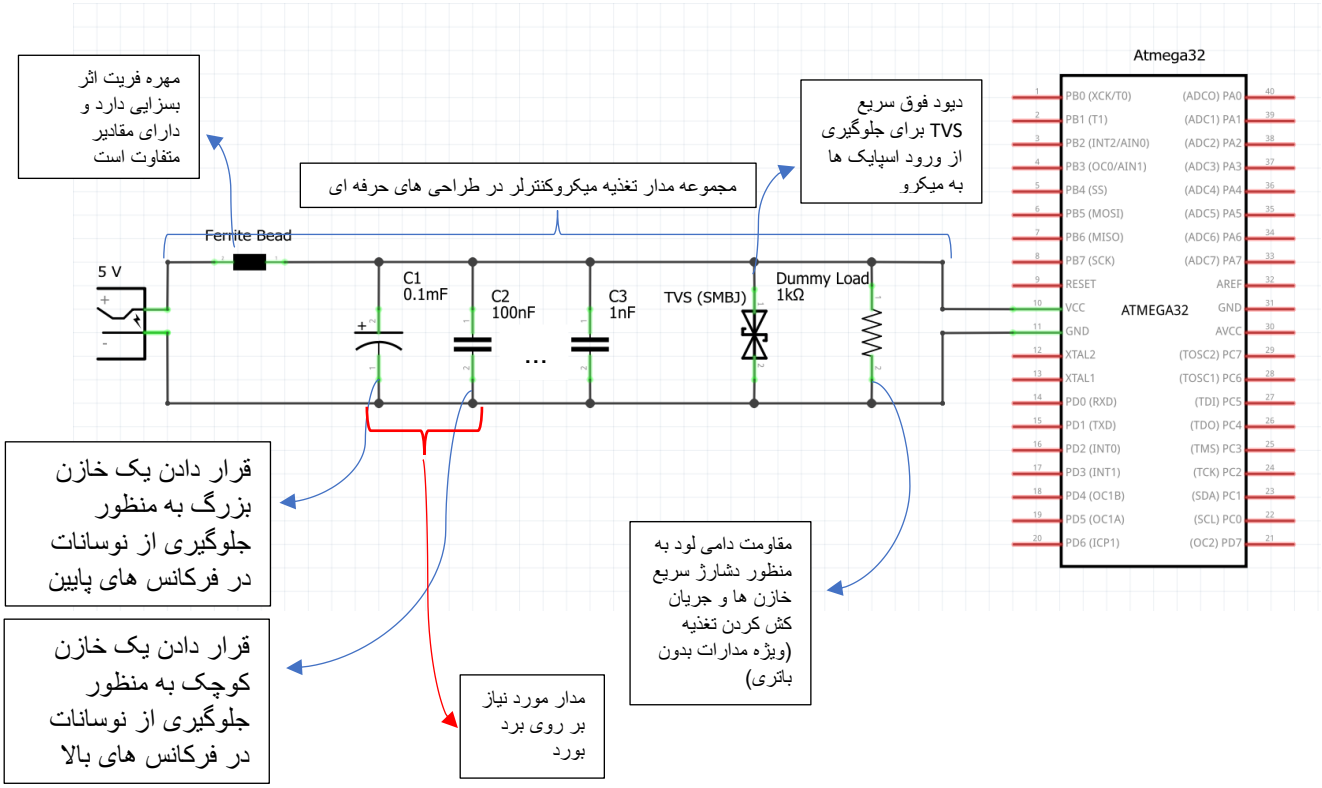
پورت ها و نمای میکروکنترلر ATMega32 :



راه اندازی :

1- تغذیه :





ضربه یا اسپایک هر چقدر هم کوچک باشد موجب هنگ کردن و نقص در سیستم شما هنگام کارکردن میشود.

نام دیگر خازن ها خازن های بایپس یا جبران ساز میگوییم.

دیود های TVS رفتاری به مانند دیودهای زنر دارند و در دو مدل یک طرفه و دو طرفه می باشد. از مدل دو طرفه در زمان هایی که ولتاژ منفی داریم استفاده میکنیم. این دیودها بسیار سریع هستند (Ultra Fast) و میتواند سریع ترین اسپایک ها را هم بگیرد. این دیود ها ضربه ها را بر روی آستانه ولتاژ خود قطع میکنند.

مهره فریت نیز اثر بسزایی در مدار ها دارد. و دارای مقادیر مختلف می باشد . و مانند مدار معادل LC عمل میکند و اثر سلفی دارد. مهره فریت به مانند یک سیم پیچ است که اطراف یک سیم بسته شده است. و حالت ها نامتعارف را از بین میبرد و حالت ههای ماندگار را عبور میدهد.

مقاومت دامی لود : این مقاومت بر روی تغذیه قرار میگیرد و به صورتی محاسبه میشود که از خود بتواند حدودا 50 میلی آمپر از خودش عبور میدهد.

استفاده از مقاومت دامی لود در مدارهایی که با باتری کار میکنند استفاده نمیشود.

مقاومت های دامی لود 2 کار مهم را انجام میدهد.

1- تغذیه را جریان کش میکند
2- خازن های موجود بر روی مدار پس از قطع تغذیه به سرعت تخلیه میشود و تغذیه به سرعت قطع میشود.

ویژگی های کارکردن در فرکانس های بالا :

1- سرعت پردازش میکرو بالا میرود
2- اسیلاتور هم نیاز به جریان کشی دارد و جریان کشی باعث کاهش نویز پذیری میشود.

پایه ریست :

کلیدی ترین و مهم ترین پایه در کلیه میکروکنترلرها می باشد که دارای شرایط خاص است و باید برای آن تمهیداتی در نظر گرفته شود.

در دنیای دیجیتال ما دو نوع مدار خواهیم داشت :

1- مدارهای ترکیبی
2- مدارهای ترتیبی

مدارهای ترکیبی :

مداراتی که خروجی آن ها به ورودی آن ها وابسته می باشد مدارات ترکیبی میگوییم. مانند مدار های گیت

مدارات ترتیبی :

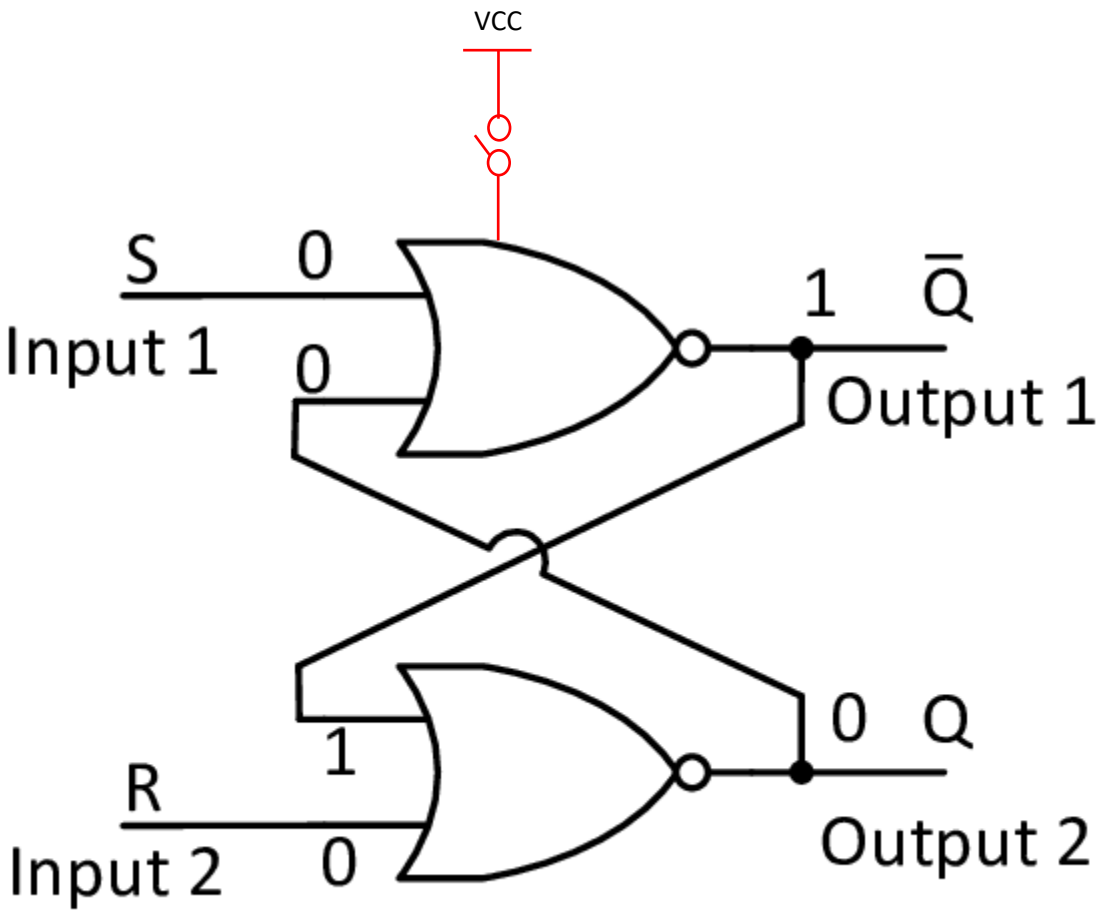
مداراتی که خروجی آن نه تنها به ورودی آن وابسته است بلکه به حالت قبل مدار هم وابستگی دارد. مانند کانترها

هسته اصلی مدارات ترتیبی را فلیپ فلاپ ها تشکیل میدهند که بنیان اصلی فلیپ فلاپ ها هم فلیپ فلاپ Latch SR می باشد.

فلیپ فلاپ Latch SR :

این فلیپ فلاپ ها را میتوان با کمک گیت های NOR یا NAND طراحی کرد

فلیپ فلاپ Latch SR کوچک ترین هسته یک فلیپ فلاپ است.



عمده ترین مشکل در این فلیپ فلاپ ها در این است که به محض متصل شدن تغذیه و روشن شدن فلیپ فلاپ بدلیل نداشتن مقدار اولیه وضعیت خروجی هایمان نیز نامشخص میماند و حالت خروجی را نمیدانیم که این مشکل عمده ترین مشکل در این فلیپ فلاپ ها می باشد.

برای حل این مشکل شرایطی را بوجود آورده ایم با نام POR (Power on Reset) مینامیم که POR از واحد هایی ساخته میشود یا به صورت سخت افزاری از داخل خود میکرو طراحی میشود. که کار این مدار ها این است که به محض روشن شدن میکرو برای مدار یک شرایط اولیه ای را تعیین میکند.

مدار POR چیست؟

این مدارها به گونه ای طراحی میشوند که پس از چند کلاک یک بار میکرو را ریست کرده و از مدار خارج میشود. که این سخت افزار ممکن است در برخی از میکرو ها وجود نداشه باشد.

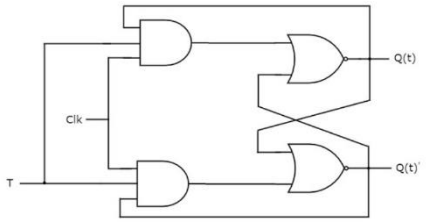
برای ریست شدن در این فلیپ فلاپ ها این را میدانیم که باید به پایه R مقدار 1 و به پایه S مقدار صفر را بدهیم تا خروجی ما که همان Q است برابر صفر و مقدار Q~ مقدار 1 را به خود میگیرد.

تمامی فلیپ فلاپ های موجود مانند D , T تماما از فلیپ فلاپ های JK تشکیل شده اند که هسته اصلی فلیپ فلاپ های JK همان Latch SR است.

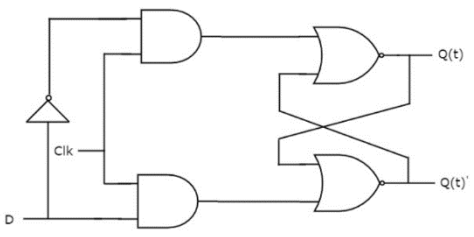
دلیل ساخته شدن فلیپ فلاپ های JK در این است که اگر ما به دو ورودی SR هر دو را 1 بدهیم خروجی را نمیتوانیم تعیین کنیم و به مشکل میخوریم. به همین دلیل Jk ها ساخته شدند و این مشکل را برطرف نمودند و خروجی را برای ما تاگل میکنند که فلیپ فلاپ های نوع T همان فلیپ فلاپ های JK می باشد که دو وردی JK به یکدیگر متصل شده و فلیپ فلاپ های T را ساخته اند.

فلیپ فلاپ های JK خروجی را در صورت دو وردی یکسان به حالت NOT حالت قبل خود در می آورند.

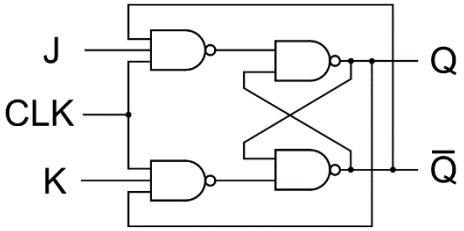
T FlipFlop



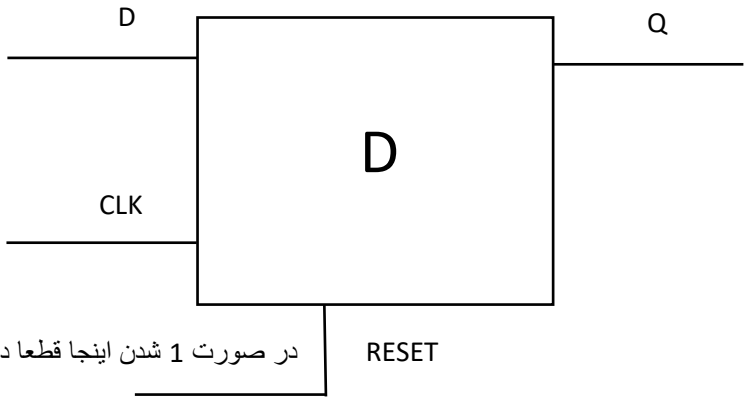
D FlipFlop



JK FlipFlop



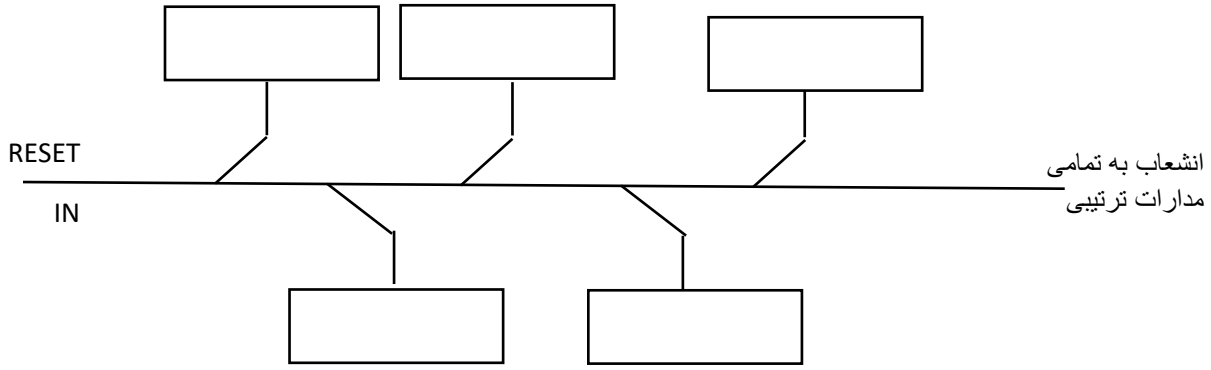
تفاوت در فلیپ فلاپ های نوع D هم در ورودی آن ها با T است که ورودی یک NOT دارد.



کار پایه ریست در این است که به سخت افزارهای ترتیبی متصل میشود و همه آنها را به مقدار اولیه خود بر میگرداند.

دلیل از ابتدا شدن برنامه در این است که در داخل میکرو یک کانتر با نام پروگرم کانتر وجود دارد. وظیفه این واحد این است که با کلاک برنامه را اجرا کند و تا انتها پیش برود. از آنجایی که کانتر یک مدار ترتیبی است و از فلیپ فلاپ T ساخته شده است و خود T ازJK ها ساخته شده است و در درون JK از SR استفاده شده است به محض فشردن ریست مقدار خروجی هم صفر میشود و کانتر هم صفر شده و از ابتدای تابع main شروع به اجرای کد میکند.

شمای داخلی اتصال پایه ریست :

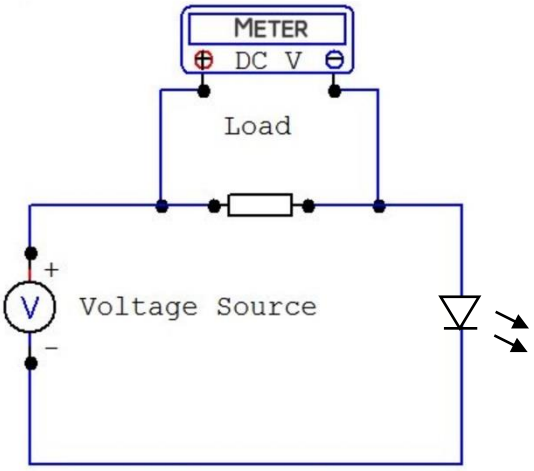


یکی دیگر از مشکلات پایه ریست بحث در ورودی بودن آن است.

به محلی که ورودی داریم Hlz میگوییم.

مفهموم HIz :

مدار زیر را در نظر داشته باشید.



ابتدا مدار را بدون مقاومت در نظر بگیرید مقدار خوانده شده توسط ولتمتر برابر 5 ولت یا همان مقدار منبع است

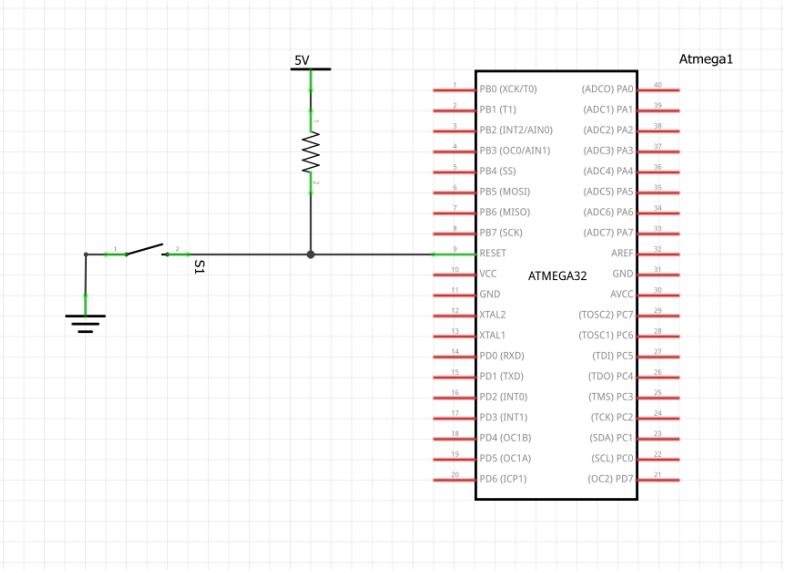
حال اگر مقاومت را با مقدار مشخص شده قرار دهیم چگونه میشود؟

اگر در مدار با وجود مقاومت یه LED هم قرار دهیم چه میشود؟

اگر وسیله اندازه گیری ما جریان کشی داشته باشد دیگر آن وسیله به کار نمی آید و یک نوع مصرف کننده خواهد بود وسیله اندازه گیری باید مقدار را فقط بخواند نه آن که جریان از خود عبور دهد یا جریان کشی کند.

به این مفهوم HIz میگویند

حال برای برطرف کردن HIz پایه ریست به این صورت عمل میکنیم.



به منظور معطوف کردن دید در HIz به یک محل به آن مقدار مورد نیازش را میدهیم تا از روی تحریکات دیگر تاثیر نپذیرد و به محض متصل شدن کلید مقدار بر روی پایه به 0 تبدیل شده و میکروی ما ریست خواهد شد.

به مدار بسته شده بر روی پایه ریست مدار پول آپ میگوییم مدارهای پول مدارهایی هستند که عکس تحریک پایه عمل میکنند.

میزان جریان در مدار پول باید کنترل شده باشد و مقدارهای مناسب برای مقاومت مقایدر زیر است

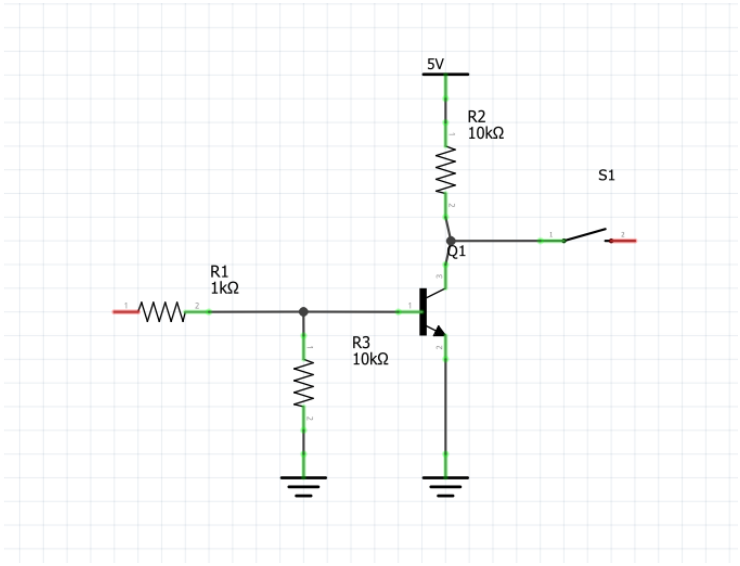
10K – 6.8K – 4.7K – 3.3K بهترین مقدار ها می باشد.

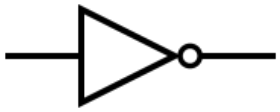
تامین کلاک	اسیلاتور خارجی
	اسیلاتور داخلی با رزوناتور کریستالی داخلی
	اسیلاتور داخلی با رزوناتور RC داخلی

ساده ترین نوع تقویت استفاده از گیت نات است به صورت تصویر

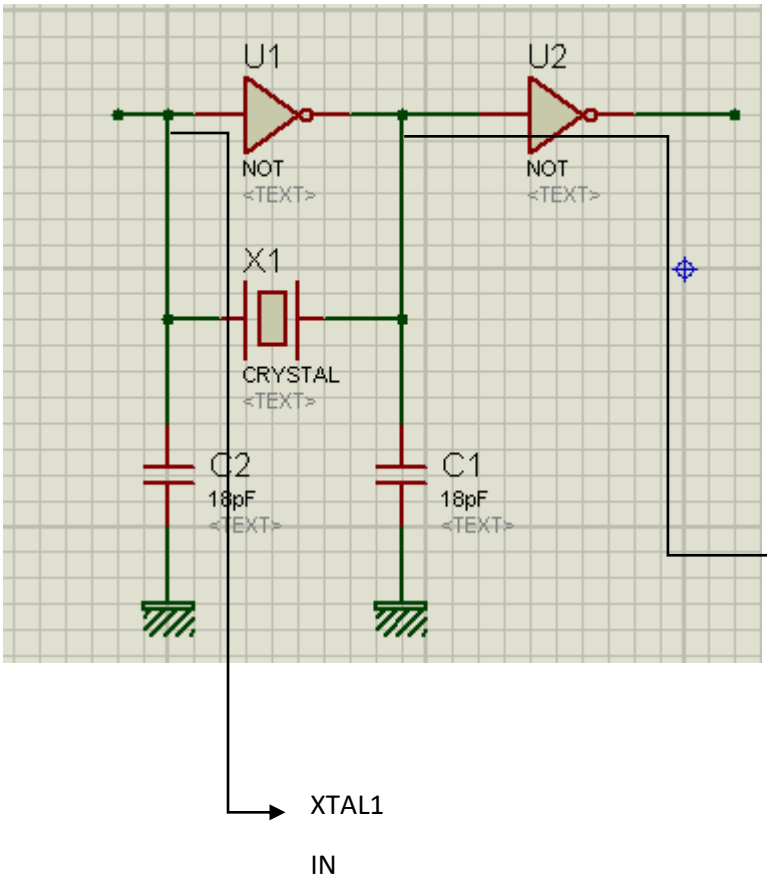
وقتی ترانزیستور خاموش است خروجی مقدار 1 و وقتی ترانزیستور فعال است خروجی 0 است

این مدار ساده ترین نوع مدار تقویت است





حال میاییم دو گیت نات را به صورت سری پشت هم قرار میدهیم و به نات اولی عنصری حساس به ضربه یا لرزش که برای تولید ولتاژ کند قرار میدهیم و یا اگر به آن ولتاژ دهیم برای ما ولتاژ را مرتعش میکند که این قطعه یک تکه تراشه کوارتز می باشد



دلیل قرار دادن نات دوم برای این است که یک مرحله هم عمل تقویت انجام شود

با نگاه کردن به پایه های میکروکنترلر دو پایه XTAL1 و XTAL2 را خواهیم دید که همان پایه های کلاک ما هستند

البته بهتر است بجای استفاده از XTAL آن ها را با نام های OSC_IN و OSC_OUT بکار ببریم

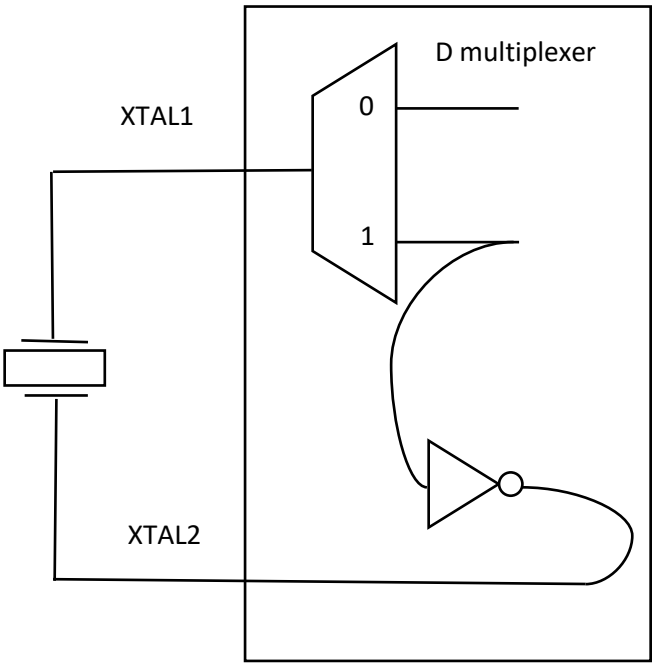
همانطور که در مدار بالا میبینید در کنار پایه های کریستال دو عدد خازن هم وجود دارد دلیل استفاده از این خازن ها در استیبل کردن و یا رزونیت کردن است که مقدار این خازن ها بسیار مهم میباشد که از 12 تا 33 پیکوفاراد میتوانید انتخاب کنید که در ای بازه بهترین مقدار از 18 تا 27 پیکوفاراد میباشد.

اگر این مقدارها بیشتر شود خازن کاری را انجام میدهد که در گرفتن نویز انجام میدهد و خروجی را صاف میکند.

نام این تکه تراشه کوارتز کریستال نام دارد که شرایط بسیار دقیق و مناسبی را برای ما ایجاد میکند.

در پشت این دو پایه در داخل میکروکنترلر یک عدد D multiplexer قرار گرفته است وظیفه این قطعه این است که تعیین کند چشمه کلاک از چه طریقی به میکرو میرسد و این قطعه حالت های انتخابی اش از طریق واحدهایی به نام فیوزبیت انجام میشود.

مانند تصویر زیر:

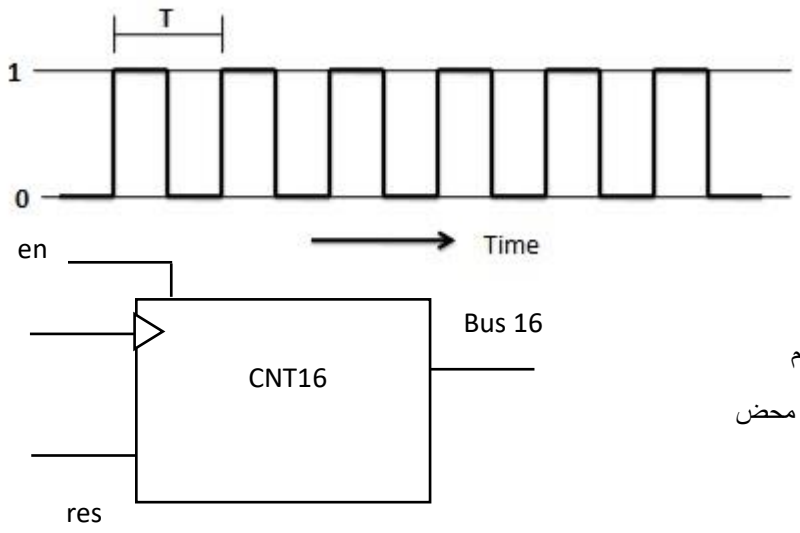


این مدار یک نمونه از نوع کریستال خارجی می باشد

اما اگر بخواهیم از یک چشمه خارجی کلاک استفاده نماییم کافی است تنها XTAL1 را به چشمه کلاک متصل نماییم و نیازی به استفاده از XTAL2 نیست

کلاک در سیستم ها به مانند یک خط کش میماند یعنی توسط کلاک اندازه گیری انجام میشود

چگونه از طریق کلاک یک زمان را اندازه گیری کنیم؟



زمانی که نیاز داریم یک زمان را از طریق کلاک اندازه گیری نماییم باید با استفاده از لبه ها این کار انجام شود فرض بر این است که پایه ریست فعال شده و مقدار ها همگی از صفر شروع میشود در کانتر با دیده شدن لبه بالا رونده یک مقدار به میزان en افزوده میشود و تا زمانی که لبه پایین را نبینیم کانتر خاموش نمیشود و عمل شمردن انجام میگیرد و به محض دیده شدن لبه پایین رونده کانتر خاموش میشود.
حال فرض کنید کانتر ما تا مقدار 30 شمرده است

این بدان معنا است که 30 لبه شمره ده شده است که هر لبه به اندازه T کلاک با لبه بعدی فاصله داردبه طور مثال اگر فرکانس ما 1 مگاهرتز باشد مقدار T ما برابر 1 میکرو ثانیه میشود و از آنجا که 30 لبه را شمردیم مقدار زمان اندازه گیری شده برابر 30 میکرو ثانیه میشود

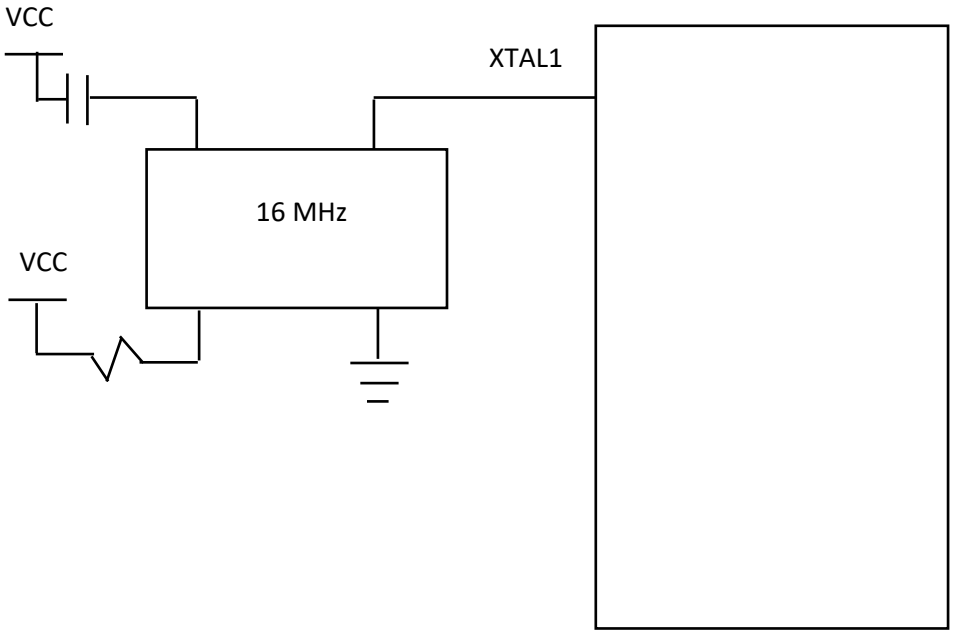
این شکل نشان میدهد که که اگر کلاک ما به درستی تنظیم نشده باشد و یا اصطلاحا جیتر داشته باشد ما نمیتوانیم زمان را به درستی اندازه گیری کنیم.

اینجاست که اهمییت کلاک بسیار معلوم میشود به صورتی که در fpga ها مسیرهای تامین کلاک دارای مسیرهای خاص است و در بیشتر موارد از جنس طلا میباشد.

کلاک ها نباید در مسیر خود به گیت ها برخورد داشته باشند زیرا گیت ها دارای تاخیر انتشار هستند و هنگام ورود به کانترها دارا اختلاف زمانی Δt می شود در طراحی ها همواره چشمه کلاک و یا کریستال ها را در نزدیک ترین مکان ممکن قرار دهند

تامین کلاک از طریق اسیلاتور خارجی :

اسیلاتور خارجی یک المان اکتیو است که جریان مصرف میکند و دارای 4 پایه است و برای کارهای دقیق استفاده میشود و یک واحد تولید کلاک محسوب میشود.



به خازنی که در کنار پایه های تغذیه قرار میگیرد خازن دیکوپلینگ میگویند که این خازن ها در نزدیک ترین نقطه به تغذیه متصل میشوند زیرا مهم است که در نزدیک ترین جای ممکن صاف ترین سیگنال را به میکرو برسانیم این خازن ها همان خازن هایی است که در زیر تمامی CPU ها قرار میگیرد

دلیل استفاده از این خازن ها در این است که اثر سلفی ترک ها را دمپ کنیم و به ازای هر پایه تغذیه یک عدد خازن با مقدار 100 نانو فاراد قرار میدهیم

در قطعه اسیلاتور یک پایه en داریم که این پایه ورودی است که یا باید آن را رها کنیم و یا پول آپ کنیم اما از آنجایی که هیچ گاه ورودی ها را رها نمیکنیم آن را پول آپ میکنیم که اگر مقدار en برابر صفر شود خروجی قطع شده و یا اگر 1 باشد خروجی وصل خواهد بود

خروجی یک اسیلاتور به صورت موج سینوسی میباشد در داخل میکروکنترلر قطعه ای به نام اشمیت تریگر قرار گرفته است که این قطعه وظیفه این را دارد که موج ها را به پالس مربعی تبدیل کند که از یک مقدار معین را 1 و از یک مقدار معین را صفر میکند

فیوزبیت ها :

همانطور که در ابتدای این جزوه امده است در میکروکنترلر حافظه ای وجود دارد به نام فلش در داخل فلش ناحیه ای وجود دارد به نام فیوزبیت که دارای مقادیر 8 یا 16 یا 32 بیت است

از آنجایی که تنها وسیله ای که میتواند حافظه فلش را تغییر دهد پروگرمر می باشد پس پروگرمر وظیفه تغییر مقدار فیوزبیت ها را دارا میباشد

پروگرمر از طریق نرم افزار و ابزار های گوناگون این فیوزبیت ها را مقدار دهی میکند.

در میکروکنترلر ATmega32 ما 16 فیوزبیت داریم که در این دوره به 5 عدد از آن ها نیاز خواهیم داشت که مهم ترین فیوز بیت ها در میکروکنترلر AVR فیوزبیت های CKSEL0 تا CKSEL3 می باشد که وظیفه این 4 فیوزبیت تعیین مسیر کلاک در میکروکنترلر می باشد

و یک فیوزبیت با نام JTAGEN را داریم که در ادامه توضیح خواهیم داد

حالت فعال در فیوزبیت ها مقدار صفر است و در حالت غیرفعال مقدار 1 می باشد

برای مشخص کردن تامین کلاک از طریق اسیلاتور خارجی می بایست هر 4 فیوزبیت CKSEL را برابر صفر قرار دهیم.

اسیلاتور داخلی با رزوناتور کریستالی خارجی :

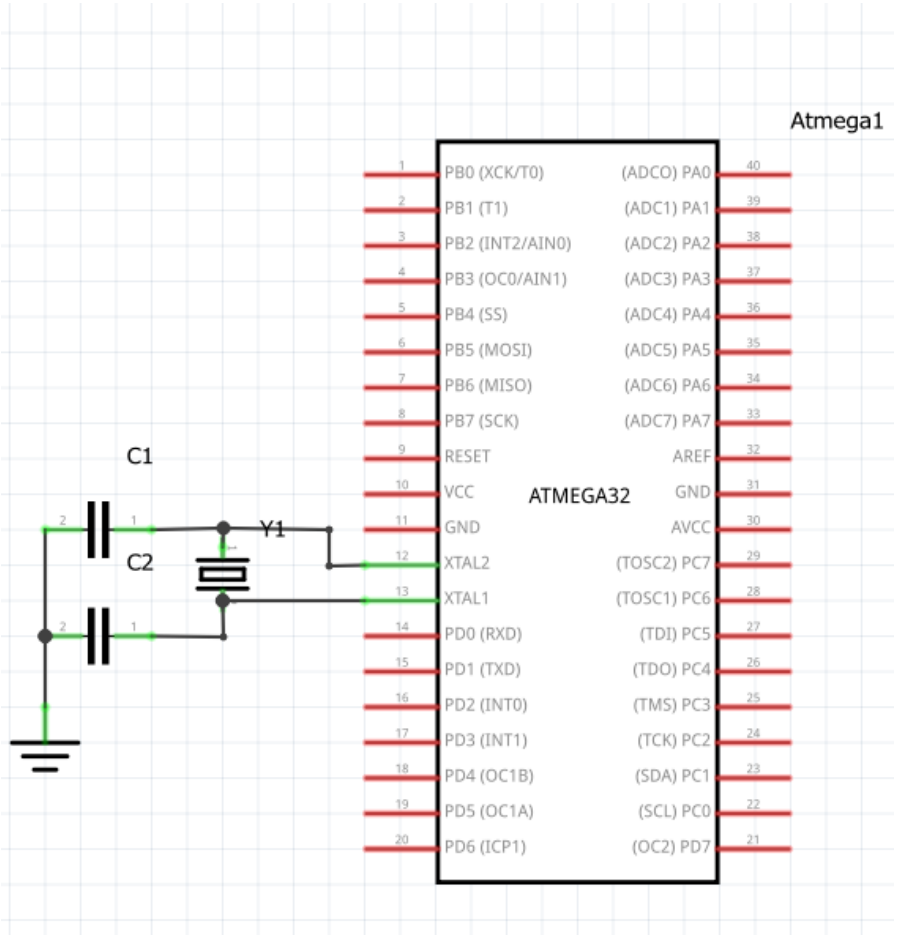
95 درصد پروژه های حال حاضر از این طریق تامین کلاک میشود

در این بخش ما به هردو پایه xtal1 و xtal2 نیازمندیم زیرا داریم از فیدبک استفاده مینماییم

در تامین کلاک از طریق اسیلاتور خارجی به xtal1 فقط نیاز بود

میزان حداکثر کریستال استفاده شده تا MHz16 می باشدیکسری کلاک های خاصی را داریم که در طراحی و حذف نویز کمک بسیار زیادی را انجام میدهد

در طراحی های حرفه ای می بایست مجموعه کریستال و خازن ها در کمترین فاصله به یکدیگر متصل شوند کریستال اخرین قطعه ایست که در مونتاژ نصب میشود و در دمای بالا اسیب میبیند و با سرعت بسیار بالایی نصب میشود.



این کریستال ها و اسیلاتور ها دارای بدنه ای فلزی هستند اما اسیلاتور ها از داخل بدنه شان گراند شده است اما کریستال ها این ویژگی را ندارند و میتوانیم با تکنیک هایی بدنه کریستال را گراند کنیم

کریستال همیشه بر روی برد نصب میشود و در زیر برد هیچگاه متصل نمیشود زیرا گرانش هم بر روی عملکرد این قطعه تاثیر گذار است.

مزیت گراند کردن بدنه کریستال

اولین مزیت این است که چون کریستال به لرزش و حرکت بسیار حساس است این روش سبب جلوگیری از لرزش آن میشود

دومین دلیل هم گراند شدن بدنه آن می باشد که تاثیر بسیار بالایی در کار و استیبل شدن کار دارد.همانند قبل برای اینکه بخواهیم کلاک را از روش اسیلاتور داخلی با رزوناتور کریستالی خارجی راه اندازی کنیم می بایست به سراغ فیوز بیت ها بریم

برای اینکه مسیر کلاک تامین شود کافی است که تمامی فیوزبیت ها را 1 قرار بدهیم

اسیلاتور داخلی با رزوناتور RC داخلی

این روش به عنوان بی کیفیت ترین روش موجود در تامین کلاک استفاده میشودو برای کارهای تایمینگ با کیفیت تاثیر خوبی ندارد و تایمینگ در برقراری ارتباط بسیار مهم است

این روش فقط برای راه اندازی اولیه و روشن شدن میکرو کاربرد دارد و نیاز به هیچ سخت افزار خارجی ندارد

این روش دارای 4 حالت می باشد

1- 8 مگاهرتز داخلی
2- 4 مگاهرتز داخلی
3- 2 مگاهرتز داخلی
4- 1 مگاهرتز داخلی

برای فعال سازی هر یک از این حالت ها به سراغ فیوزبیت ها میرویم به ترتیب شماره گذاری فیوزبیت ها به صورت زیر است

1- 0100
2- 0011
3- 0010
4- 0001

به صورت پیش فرض وقتی تراشه را خریداری میکنیم بر روی 1 مگاهرتز داخلی تنظیم شده است

بررسی پایه های AREF , AVCC , AGND

در طراحی ها هنگامی که میکسی از سیگنال های دیجیتال و انالوگ را داریم دچار بحران میشویم

در سیستم های دیجیتال اگر بر روی سیگنال ما نویز باشد جای نگرانی وجود ندارد زیرا عنصر دیجیتالی ما فقط به دو سطح 0 و 1 حساسیت دارد و مشکلی در اجرا پیش نمی اورد

اما هرگاه بر روی سیگنال های انالوگ ما دچار ضربه یا نویز شویم به دلیل اینکه سیگنال های انالوگ به صورت سیگنال های پیوسته بوده و سطح منطقی ندارد نویز هم جزیی از سیگنال ما میشود و باعث خطا در روند اجرا کار می میشود

هنگامی که این دو سیگنال را داریم این دو بر هم اثر میگذارند و اثر را هم بر روی تغذیه یکدیگر میگذارند و وقتی تغذیه به مشکل بخورد سیستم بهم میریزد برای اینکه این مشکل را نداشته باشیم می بایست بین سیگنال های اناوگ و دیجیتال فیلتر هایی را طراحی کنیم تا این اثرات بر روی انالوگ دمپ شود

به عناصری که از آن ها یا به آن ها سیگنال انالوگ وارد یا خارج میشود عنصر انالوگی میگوییم و در غیر این صورت عنصر دیجیتالی است

عناصر دیجیتالی در برابر نویز مستحکم هستند اما عناصر انالوگی در برابر نویز ها بسیار حساس هستند

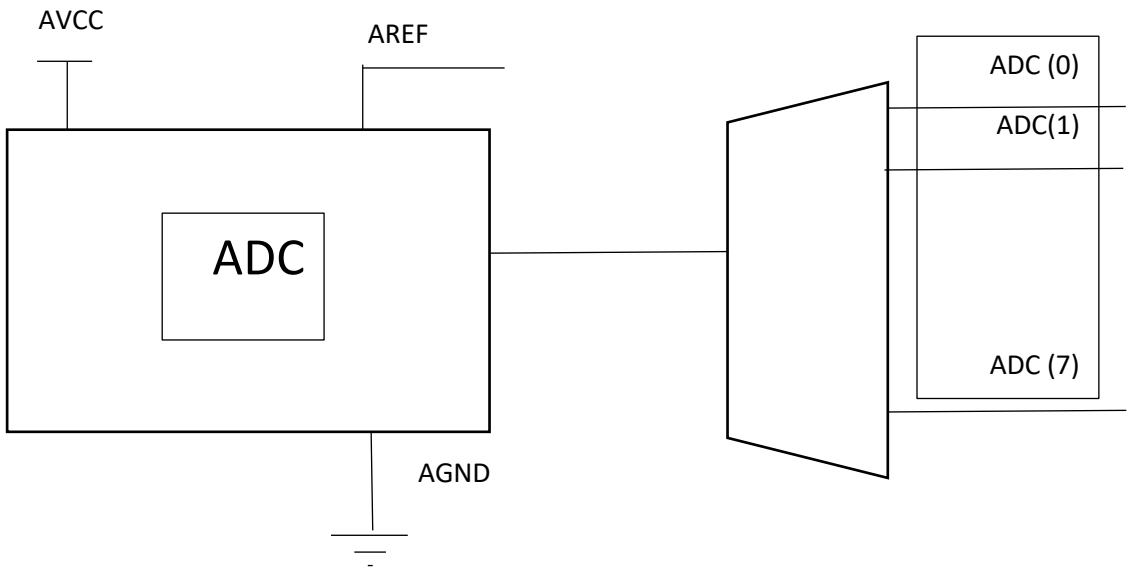
به صورتی که طراحان تغذیه این دو عناصر را از یکدیگر جدا میکنندر طراحی ها سعی میشود که عناصر انالوگی در یک بلوک و کنار هم و عناصر دیجیتالی هم در یک بلوک و کنار هم قرار بگیرند و بین این بلوک ها فاصله ای را در نظر میگیرند

برای اینکه ما دوتا ورودی تغذیه در مدارهایمان نداشته باشیم هنگام راه اندازی میاییم از یک فیلتر پایین گذر LC استفاده میکنیم برای VCC و AVCC اما برای گراند های کافی است از یک مهره فریت استفاده شود

این اثر بقدری چشم گیر است که دو پایه گراند مجزا در روی میکرو ها در نظر میگیرند

با توجه به میکروکنترل ATmega32 ما یک واحد 8 بیتی انالوگ به دیجیتال را میبینیم که این واحر با کمک یک مالتی پلکسر 8 بیت به یک مبدل انالوگ به دیجیتال متصل شده است

این مالتی پلکسر میتواند در هر لحظه یکی از ورودی ها را بخواند و تبدیل کند بر روی این مبدل یک پایه با نام AREF وجود دارد که مخفف Analog Voltage Reference که می توانیم با کمک این پایه میزان سطح خوانایی را مشخص کنیم این پایه این کمک را به ما میکند که با کمک رزولوشن و مقدار آن مقدار خواندن انالوگ را بسیار دقیق انجام دهیم



اگر به محل قرار گیری پایه های میکرو توجه کنیم میبینیم که واحد ADC به همراه پایه های تغذیه و رفرنس در یک سمت واحد و کنار هم قرار گرفته اند و این مدار ها به صورت کلی رعایت میشود

برنامه ریزی :

میکروکنترلرهای AVR به دو صورت برنامه ریزی میشوند

1 – روش SPI

2 – روش JTAG

هر دو مورد از پروتکل های انتقال به صورت سریال میباشند اما SPI برای کار های ارتباطی دیگر هم مورد استفاده قرار میگیرد

پروتکل SPI (serial peripheral interface) یک پروتکل سنکرون برای انتقال می باشد

JTAG : (join test action group) یک پروتکل انتقال داده که عموما برای برنامه ریزی سیستم ها مورد استفاده قرار میگیرد و به صورت مرسوم مورد انتقال داده با عناصر و المان ها قرار نمیگیرد.

تمامی ARM ها و FPGA ها از این طریق برنامه ریزی میشوند

اما به دلیل هزینه بالای پروگرمرهای JTAG در AVR مورد استقبال قرار نگرفت و از SPI استفاده شد

پروگرمرهای MK II از طریق ارتباط JTAG میکرو ها را پروگرم میکردندو یکی از مزیت های JTAG در پروگرمر ها امکان دیباگینگ به صورت سخت افزاری بودبه صورتی که میکرو کنترلر از یک سمت به کامپیوتر و از سمت دیگر به میکروکنترلر ما متصل میشود و میتوانید به صورت همزمان کد های خود را از هرجایی که میخواهید اجرا کنید و تمامی تغییرات را به صورت زنده مشاهده کنید

از آنجایی که ابزار های دیباگ یا دیباگر ها ابتدا برنامه رو پروگرم میکنند و بعد عمل دیباگ را انجام میدهند برای برنامه ریزی نیز مورد استفاده قرار میگیرند

پایه های مرتبط با ارتباط به صورت JTAG به صورت زیر میباشد :

TDI : جهت زنجیره سازی در JTAG

TDO : جهت زنجیره سازی در JTAG

TMS : دیتا

TCK : کلاک

به جز 4 پایه اصلی بالا در هر میکرو کنترلر به صورت مجزا پایه های دیگری به JTAG اختصاص میدهند که در میکروکنترلر ATMega32 پایه های RESET , VCC , GND نیز مورد استفاده قرار میگیرد

هنگامی که ما یک میکروکنترلر نو خریداری میکنیم واحد JTAG به صورت پیش فرض برای ما فعال شده است جهت غیرفعال سازی این واحد با استفاده از فیوزبیت ها این واحد را غیرفعال میکنیم

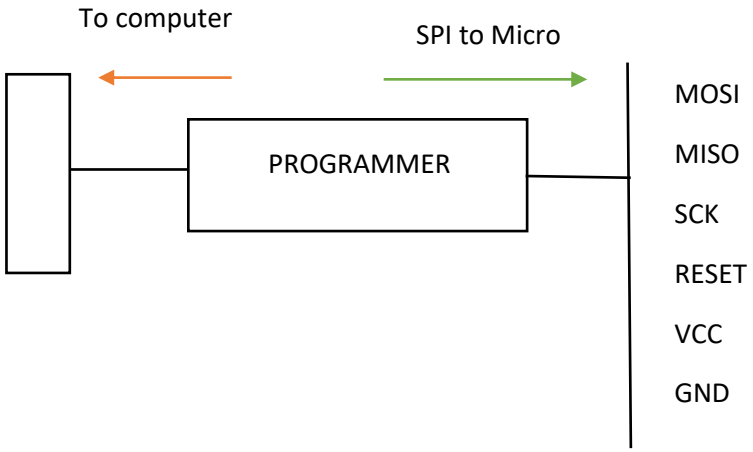
فیوزبیت JTAGEN به همین منظور در میکروکنترلر ما قرار گرفته استو جهت غیرفعال کردن این فیوزبیت مقدار آن را 1 میگذاریم.

SPI :

تمامی پروگرمر های امروزه ما از طریق ارتباط SPI عمل برنامه ریزی را انجام میدهند که این پروگرمر ها نیز استاندارد هایی دارند

Stk200/300 : در اصل یک ارتباط با کامپیوتر است که به صورا ارتباط پاراال عمل میکند که ار اصل یک نوع بافر است بافری که سیستم شما را از میکرو ایزوله میکند. و جلو صدمه دیدن سیستم شما را در اثر یک خطا و اشتباه محافظت میکند که این ارتباط از یک سو پارال و از سوی دیگر SPI است.

Stk500 : یک پروگرمر ارتباط سریال است که با کمک RS232 این ارتباط را برقرار میکند این پروگرمر ها از یک سو ارتباط سریال و از سمت دیگر ارتباط SPI دارند



ارتباط میکروکنترلر جهت برنامه ریزی با پروگرمر به صورت تصویر بالا انجام میشود

پروتکل SPI هم دارای 4 پایه اصلی جهت برقراری ارتباط می باشد که به صورت زیر است :

Master Output Slave Input : MOSI

Master Input Slave Output : MISO

کلاک : SCK

Slave Select : SS

جهت برنامه ریزی از 3 پایه اصلی پروتکل SPI استفاده میشود و از پایه SS جهت برنامه ریزی استفاده نمیکنند پس از برقراری ارتباط با میکروکنترلر باقی عملیت برنامه ریزی را نرم افزار پروگرمر انجام میدهیم.

دقت کنید در تمامی خانواده های AVR به جز ATMega2560 , ATMega128 , ATMega64 پایه های MISO , MOSI جهت برنامه ریزی میباشند در این 3 میکروکنترلر پایه های PDI و PDO برای برنامه ریزی استفاده میشوند و پایه های MISO , MOSI بر روی میکرکنترلر فقط برای انتقال داده مورد استفاده قرار میگیرد

PDI: معادل MOSI

PDO : معادل MISO

و پایه ای RESET , GND , VCC همان پایه ها می باشند و تفاوتی ندارد

GPIO

INPUT

OUTPUT

در این بخش OUTPUT را بررسی خواهیم کرد

در میکرو کنترلر AVR ما 4 پورت داریم که هر پورت 8 پین دارد

برای کار کردن با هر پورت ما 3 عدد رجیستر داریم : DDR , PORT , PIN

رجیستر DDR :

این رجیستر جهت را در پورت ها مشخص میکند وظیفه این رجیستر در این است که مشخص کند پورت ما ورودی شود و یا خروجی جهت خروجی کردن پورت می بایست مقدار این رجیستر در هر خانه مورد نظر مقدار 1 قرار گیرد و جهت ورودی مقدار 0 را قرار میدهیم.

DDRx -------> DDRA

						0	1

در یک پورت ما میتوانیم هر پایه ای را که بخواهیم ورودی و یا خروجی کنیم.

جهت مشخص کردن مقدار اولیه هر پایه باید از رجیستر PORT استفاده نماییم

رجیستر پورت 2 عمل را برای ما انجام میدهد

1- تعیین مقدار اولیه هر پایه مورد نظر مقدار 1 یا 0
2- برای گرفتن ورودی ها استفاده میشود که در ورودی مورد استفاده قرار میگیرد

از آنجایی که رجیستر ها از جنس SRAM هستند نمیتوانند مقدار خود را با قطع و وصل جریان نگهدارند و باید مقدار دهی شوند و وقتی به صورت پیش فرض راه اندازی میشود مقدار 0 را دارا می باشند.

اگر رجیستر ها را مقدار دهی نکنیم پورت ما به صورت ورودی و با مقدار اولیه 0 هستند.

دلیل این کار هم برای این است که اگر به صورت ناخواسته به یک پایه ای که خروجی شده مقدار 1 دهیم (به VCC) متصل شود پورت ما اسیب نبیند.

PORTx --------> PORTA

						0	1