

بسمه تعالی - تمرین برنامه نویسی ARM - میکروپروسسور - دانشگاه صنعتی اصفهان - صدری - سری دوم -

۱- الف) توضیح دهید که کاربرد اصلی واحد PIT در میکروکنترلرهای مبتنی بر ARM چیست؟

واحد programmable interval timer برای تولید interrupt با بازه های زمانی مشخص استفاده می شود. مثلاً فرض کنید که یک task مشخصی قرار است در بازه های زمانی مشخص اجرا شود. از pit برای صدا زدن آن task می توان استفاده نمود.

ب) می خواهیم برای یک میکروکنترلر مبتنی بر ARM ساخت ATME1، یک برنامه به زبان C بنویسیم که روی یکی از پایه های خروجی میکرو، یک پالس با فرکانس 0.5Hz تولید کند. چه راه هایی برای انجام این کار داریم؟ توضیح دهید.

۱- استفاده از یک حلقه معمولی، این حلقه آنقدر دور می زند تا تاخیر مورد نظر ایجاد شود. داخل حلقه قرار نیست کاری انجام شود. صرفاً می خواهیم یک سری دستور همینطوری را اجرا کنیم تا تاخیر مورد نظر به دست بیاید. این ساده ترین روش و ضعیف ترین روش است. CPU در این روش کار دیگری نمی تواند انجام دهد و مدام در حال اجرا کردن دستورات داخل حلقه است. مصرف توان هم بالا می رود.

۲- می توانیم از یک PIT برای این کار استفاده کنیم. مثلاً PIV را، یعنی آن Threshold ای که شمارنده تا آن مقدار می شمارد و سپس دوباره صفر می شود را، به گونه ای تنظیم می کنیم که فرکانس تولید Interrupt دو برابر فرکانس مورد نیاز ما باشد. حال در interrupt service routine خروجی را، هرچه که هست not می کنیم.

۳- از واحد timer/counter هم می توان استفاده کرد.

ج) فرض کنید میکروکنترلر ما با فرکانس 48 مگاهرتز clock می خورد. می خواهیم با PIT پالس مورد نظر را تولید کنیم. مقدار ثابت PIV چقدر باید باشد؟ آیا نیاز به چک کردن مقدار PICNT خواهیم داشت یا نه؟

اگر بخواهیم فرکانس خروجی 0.5Hz باشد، interrupt service routine مربوط به PIT باید با فرکانس ۱ هرتز اجرا شود. یعنی هرثانیه یک بار اجرا شود. و در هر باری که اجرا شد، خروجی میکرو را NOT کند. به این ترتیب ما طی ۲ ثانیه یک پالس کامل خواهیم داشت و فرکانس آن نیم هرتز است.

مطابق با شکل مربوط به ساختار PIT در Datasheet می بینیم که پالس ساعتی که باعث شمردن در PIT می شود، فرکانسش 1/16 فرکانس پالس ساعت اصلی است. پس اگر پالس ساعت اصلی سیستم ۴۸ مگاهرتز است، فرکانس شمارش در PIT برابر با ۳ مگاهرتز است.

حالا اگر مقدار PIV را برابر با بزرگترین مقدار ممکن قرار دهیم (یعنی حدودا ۱ میلیون) (چون شما با عدد ۲۰ بیتی تا ۱ میلیون را می توانید نشان دهید) آنوقت، CPIV از صفر تا ۱ میلیون می شمارد، وقتی به ۱ میلیون رسید یک وقفه تولید می کند و PICNT یک واحد زیاد می شود.

پس فرکانس تولید وقفه در PIT برابر با 1/3 ثانیه خواهد بود. چون شمارنده ما با فرکانس ۳ میلیون هرتز کار می کند.

برای آنکه بتوانیم ۱ ثانیه را اندازه بگیریم، به PICNT نیاز داریم. هر بار که 1/3 ثانیه زمان می گذرد PICNT یک واحد زیاد می شود. وقتی PICNT را خواندیم و مقدار آن برابر با ۳ بود، یعنی به اندازه 3 عدد 1/3 ثانیه زمان گذشته. یعنی ۱ ثانیه زمان گذشته. پس در interrupt service routine مربوط به PIT هر بار PIT_PIR را می خوانیم. وقتی PICNT برابر با ۳ بود، پایه خروجی میکرو را NOT می کنیم. و البته بهتر است در این لحظه با خواندن از PIT_PIVR ثبات PICNT را صفر کنیم.

(د) برای این که این برنامه را بنویسیم، چه قسمت هایی از میکروکنترلر بایستی فعال شوند؟ واحد PMC کدام بخش ها را باید فعال کند؟ تکه کدی که این کار را انجام می دهد، بطور حدودی بنویسید.

(برای نوشتن کد، شما به Header File ای که مربوط به میکروکنترلر است، دسترسی دارید.)

(د) برنامه C ما، چه روتین هایی خواهد داشت؟ اسم روتین هایی که ضروری هستند را بنویسید و کارهای مهمی که در هر کدام انجام می شود را مشخص کنید.

(ه) برنامه C ای که برای interrupt service routine مربوط به PIT می نویسیم چگونه است؟ بطور حدودی آن را بنویسید.

(و) آیا در این برنامه نیاز به نوع volatile برای هیچ یک از متغیرهایمان داریم؟

۲- توضیح دهید که تکه کد زیر چه کاری انجام می دهد. (خط به خط توضیح دهید)

```
void init_pit (void) {           /* Setup PIT with Interrupt */

    AT91S_AIC * pAIC = AT91C_BASE_AIC;

    *AT91C_PITC_PIMR = AT91C_PITC_PITIEN | /* PIT Interrupt Enable */

        AT91C_PITC_PITEN | /* PIT Enable */

        PIV;           /* Periodic Interval Value */

    /* Setup System Interrupt Mode and Vector with Priority 7 and Enable it */

    pAIC->AIC_SMR[AT91C_ID_SYS] = AT91C_AIC_SRCTYPE_INT_POSITIVE_EDGE | 7;

    pAIC->AIC_SVR[AT91C_ID_SYS] = (unsigned long) pit_int;

    pAIC->AIC_IECR = (1 << AT91C_ID_SYS);

}
```

ب) هر کدام از ثبات های SMR و SVR چه کاری انجام می دهند؟

۳- می خواهیم با استفاده از درگاه سریال در میکرو داده ها را با Baud Rate ای برابر ۳۸۴۰۰ بیت بر ثانیه انتقال دهیم.

الف) اگر فرکانس کاری میکرو برابر با 3.68MHz باشد، مقدار CD چقدر باید باشد تا به Baud Rate مورد نظر برسیم؟

ب) یک روتین ساده به C برای ارسال یک کاراکتر توسط UART بنویسید.

ج) برنامه زیر چه کاری انجام می دهد؟ بطور دقیق توضیح دهید.

```
int sendchar (int ch) {

    if (ch == '\n') {

        while (!(pUSART->US_CSR & AT91C_US_TXRDY));        pUSART->US_THR = '\r';

    }

}
```

```
while (!(pUSART->US_CSR & AT91C_US_TXRDY));  
return (pUSART->US_THR = ch);  
}
```

د) اگر بخواهیم از sub-routine بالا برای دریافت داده استفاده کنیم ، چه مشکل و ایراد عمده ای ممکن است در عملکرد سیستم رخ دهد؟