



بسمه تعالی

درس طراحی سیستم‌های نهفته مبتنی بر FPGA  
آزمایش ۲: طراحی Audio Player با پردازنده NIOS II

پردیس دانشکده‌های فنی دانشگاه تهران  
دانشکده مهندسی برق و کامپیوتر  
دکتر بیژن علیزاده

دستیاران آموزشی:

siamackbm@yahoo.com	سیامک بیگ محمدی
hamid.imani74@gmail.com	حمیدرضا ایمانی
mohsenfathi7@yahoo.com	محسن فتحی

پاییز ۱۳۹۷

---

## مدت آزمایش: دو جلسه

## اهداف آزمایش:

- ✓ آشنایی با پردازنده NIOS II
- ✓ آشنایی با برنامه‌نویسی به زبان C برای پردازنده Nios II
- ✓ آشنایی با DE2 Media Computer
- ✓ آشنایی اولیه با روند طراحی سخت افزاری/نرم‌افزاری یک سیستم مبتنی بر Nios II

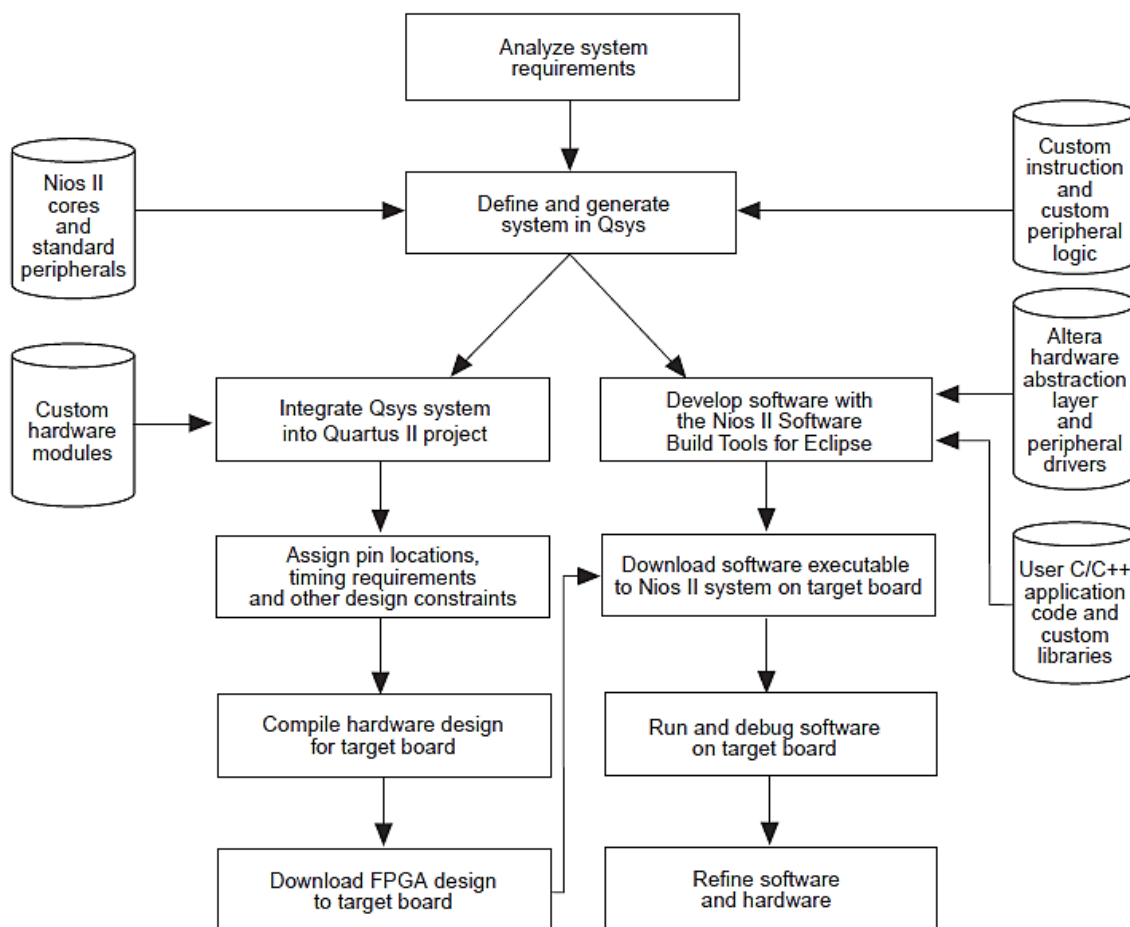
## مقدمه

پردازنده Nios II که در سه نسخه سریع (f)، استاندارد (s) و اقتصادی (e) عرضه شده است، پردازنده‌ای ۳۲ بیتی و قابل پیکربندی<sup>۱</sup> با معماری RISC برای استفاده در FPGAها و SoCهای شرکت Altera است. البته امکان استفاده از این پردازنده به صورت ASIC نیز وجود دارد. یک سیستم کامل می‌تواند با قرارگیری چندین ماژول مختلف در کنار هسته پردازنده Nios II تشکیل شود. شکل ۱ روند کلی طراحی سخت‌افزاری/نرم‌افزاری

---

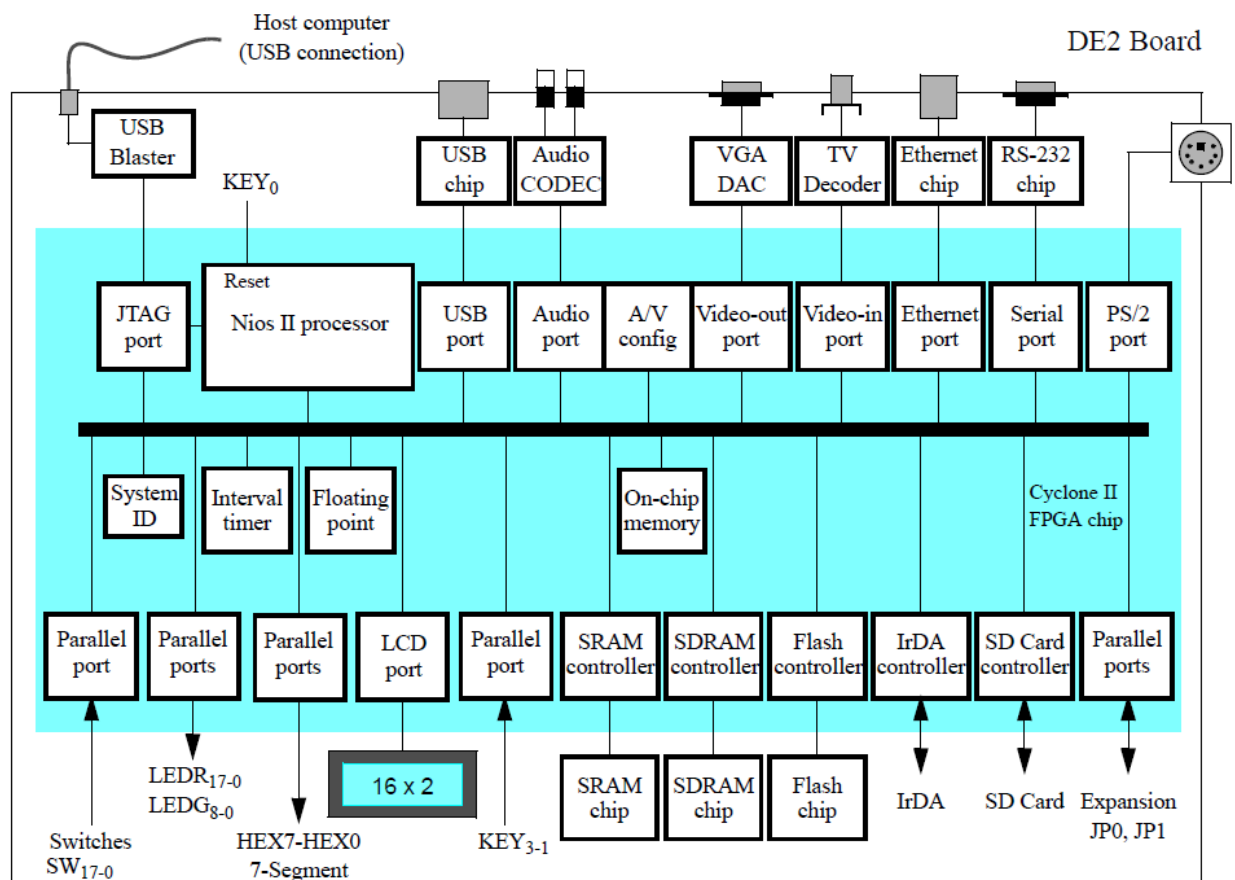
<sup>1</sup> Configurable

یک سیستم مبتنی بر پردازنده‌ی Nios II را نشان می‌دهد. این روند را به صورت کامل در گزارش خود توضیح دهید.



شکل ۱ روند طراحی سیستم مبتنی بر Nios II.

برای مثال شکل ۲ اجزای سیستم DE2 Media Computer را نشان می‌دهد که به کمک نرم‌افزار Qsys با متصل کردن IP Core های مختلف (از جمله خود هسته Nios II) به یکدیگر به کمک باس‌های استاندارد ساخته شده است. در این آزمایش به جنبه‌های نرم‌افزاری این سیستم خواهیم پرداخت و از این سیستم به عنوان بستر سخت‌افزاری آماده استفاده خواهیم کرد. برای انجام این آزمایش داشتن آشنایی با زبان C ضروری است.



شکل ۲ بلوک دیاگرام DE2 Media Computer.

## شرح آزمائش

در این آزمایش ابتدا یک سیستم سخت‌افزاری کامل از پیش طراحی شده به نام DE2 Media Computer را به کمک نرم‌افزار Quartus بر روی FPGA انتقال می‌دهیم و با یک برنامه آماده به کمک Nios II Software Build Tools for Eclipse آن را تست می‌کنیم (این کارها را می‌توان با Altera Monitor Program هم انجام داد اما جهت آشنایی با روند اصلی از دو برنامه دیگر استفاده می‌کنیم). سپس با مفهوم درایورنویسی آشنا می‌شویم و درایور ساده‌ای برای به کارگیری ماوس PS/2 در سیستم DE2 Media Computer می‌نویسیم. در نهایت یک دستگاه ضبط و پخش صدا با نمایش گرافیکی روی مانیتور و قابلیت ایجاد echo درست خواهیم کرد. ابتدا Altera Monitor Program نسخه 13.0 را نصب نمایید تا IP Coreها، داکيومنت‌ها و خود سیستم‌های کامپیوتری نمونه از جمله DE2 Media Computer نصب شوند. سپس مراحل زیر را به ترتیب انجام دهید:

## ۱- آشنایی با DE2 Media Computer و اجرای کد نمونه

به منظور آشنایی با DE2 Media Computer به فایل راهنمای این سیستم در پوشه doc در مسیر  
altera\13.0sp1\University\_Program\NiosII\_Computer\_Systems\DE2\DE2\_Media\_Computer  
مراجعه کنید. در این مسیر فایل پروژه کامل این سیستم به دو زبان وریلاگ و VHDL قرار دارد. همچنین  
برنامه‌های نمونه به زبان C با و بدون استفاده از HAL<sup>۲</sup> در این مسیر قرار گرفته‌اند. مراحل زیر را انجام دهید:

(۱) نسخه تغییر یافته‌ای از داکيومنت DE2 Media Computer که فقط شامل بخش‌هایی است که خواندن آن-  
ها برای انجام این آزمایش مفید است در سایت درس بارگذاری شده است. ابتدا این فایل را مطالعه کنید.  
همچنین فایل راهنمای کنترلر PS/2 را از پوشه doc در مسیر زیر مطالعه نمایید:

altera\13.0sp1\ip\University\_Program\Input\_Output\altera\_up\_avalon\_ps2  
همچنین بخش مربوط به Pixel Buffer DMA Controller را از فایل راهنمای IP Core های ویدئو در  
پوشه doc در مسیر زیر مطالعه نمایید. برای این آزمایش خصوصاً فهم دقیق عملکرد Back Buffer اهمیت  
دارد.

altera\13.0sp1\ip\University\_Program\Audio\_Video\Video  
علاوه بر عملکرد IP Core ها در این فایل‌ها می‌توانید با توابعی که برای استفاده از این IP Core ها در HAL  
تعریف شده‌اند نیز آشنا شوید.

(۲) پوشه Verilog مربوط به DE2 Media Computer را به مسیر دیگری کپی کنید (حتماً این کار را انجام  
دهید و فایل‌های اصلی را تغییر ندهید زیرا برای آزمایش‌های بعدی استفاده خواهند شد). برد DE2 را به  
کامپیوتر متصل کنید و سیستم DE2 Media Computer را با پروگرام کردن FPGA با فایل  
DE2\_Media\_Computer.sof به آن منتقل کنید. در این آزمایش هیچ تغییری در سخت‌افزار DE2  
Media Computer اعمال نخواهد شد.

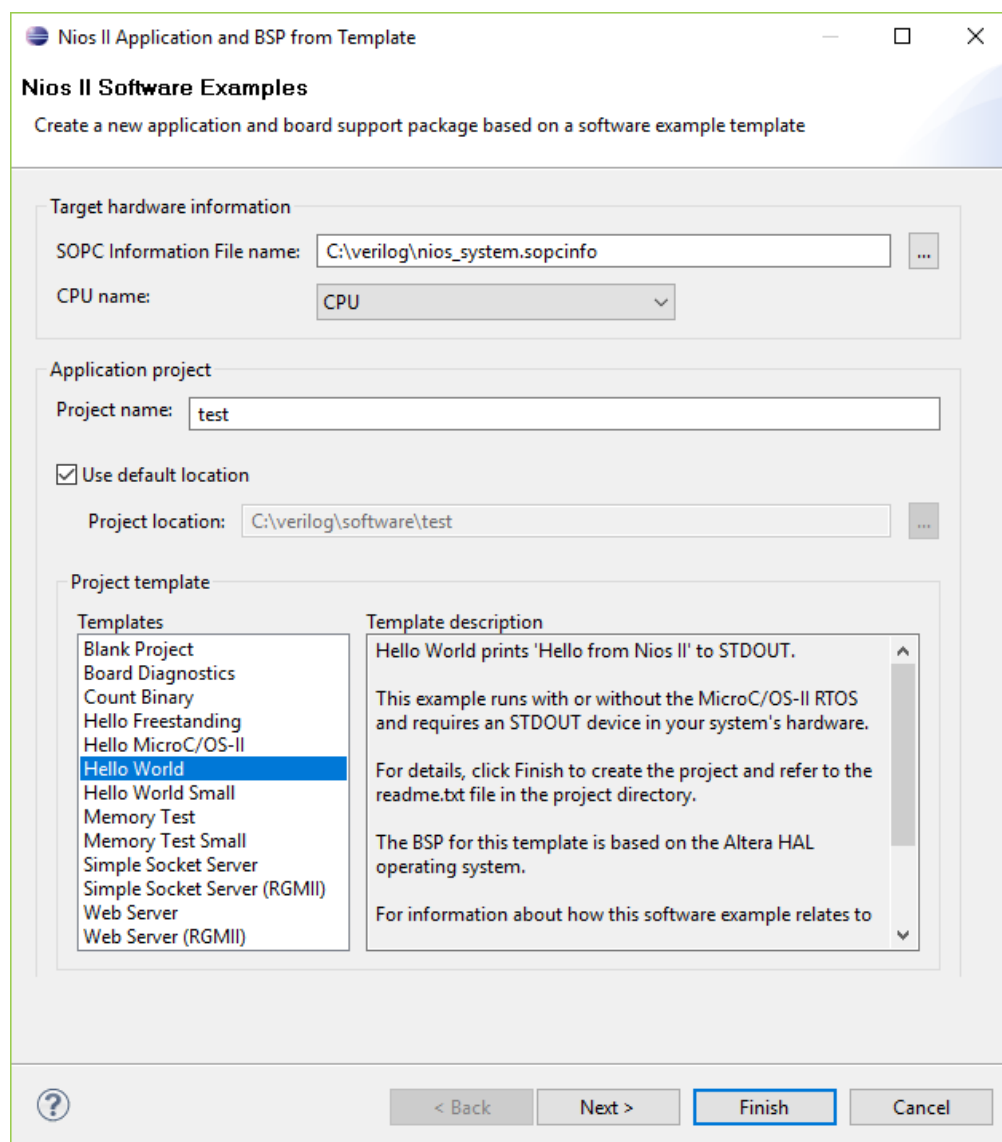
(۳) Nios II 13.0sp1 Software Build Tools for Eclipse را اجرا نمایید (این نرم‌افزار یک IDE بر مبنای  
Eclipse است که امکان کدنویسی برای پردازنده Nios II را فراهم می‌کند. نرم‌افزار به همراه Quartus II  
نصب شده است). پس از تعیین یک workspace (که اطلاعات مربوط به پروژه‌های Eclipse و مکان  
قرارگیری آن‌ها را شامل خواهد شد) محیط Eclipse را مشاهده خواهید کرد.

(۴) از منوی File گزینه New → Nios II Application and BSP from Template را انتخاب نمایید و در  
پنجره ظاهر شده مسیر فایل nios\_system.sopcinfo را که شامل اطلاعات IP Core های موجود در



---

<sup>2</sup> Hardware Abstraction Layer

سیستم، اتصالات بین آن‌ها و آدرس قرارگیری هر IP Core است، وارد نمایید. بر اساس این اطلاعات، درایورهای مورد نیاز (HAL) در قالب یک پروژه BSP (Board Support Package) ساخته خواهد شد. نام پروژه را به دلخواه انتخاب نمایید و در صورتی که می‌خواهید پروژه خود را در میسر دیگری غیر از مسیر پیش‌فرض ایجاد کنید، آن را تغییر دهید. ابتدا مثال Hello World را انتخاب نمایید. شکل ۳ تنظیمات نمونه برای ایجاد پروژه جدید را نشان می‌دهد.



شکل ۳ صفحه ایجاد پروژه جدید در Eclipse.

۵) برای اجرای عادی برنامه بر روی پروسسور Nios II، از دکمه  و برای دیباگ آن از دکمه  استفاده کنید و سپس Nios II Hardware را انتخاب کنید. همچنین می‌توانید بر روی نام پروژه در پنجره Project

Explorer راست کلیک کرده و یکی از زیرمنوهای Run As... یا Debug As... را انتخاب نمایید. در این بخش برنامه Hello World را به صورت عادی اجرا نمایید (در بخش‌های بعدی استفاده از مد دیباگ و تعریف Breakpoint ها می‌تواند راهگشا باشد). با این کار کد از طریق رابط USB Blaster و اینترفیس JTAG به مموری‌های موجود در DE2 Media Computer انتقال می‌یابد و اجرا می‌شود. اولین اجرا به دلیل نیاز به کامپایل BSP که شامل تمامی درایورهای سیستم DE2 Media Computer است، کمی با تأخیر شروع خواهد شد اما در نهایت انتظار می‌رود با اجرای برنامه Hello World که کد ساده زیر است، در پنجره Console عبارت Hello from Nios II! چاپ شود.

```
#include <stdio.h>
```

```
int main()  
{  
    printf("Hello from Nios II!\n");  
    return 0;  
}
```

۶) حال یک مانیتور به پورت VGA برد وصل نمایید. کدهای موجود در مسیر زیر را (به غیر از فایل با پسوند ncf که مربوط به Altera Monitor Program است) به پروژه کپی کنید (فایل hello\_world.c را پاک کنید) و برنامه جدید را اجرا کنید.

...\DE2\_Media\_Computer\app\_software\_HAL\media\_interrupt\_HAL

۷) کدهای مربوط به media\_interrupt\_HAL را به طور کامل مطالعه کنید و خلاصه عملکرد آن را گزارش نمایید. بدنه اصلی برنامه (تابع main) در فایل media\_interrupt\_HAL.c قرار دارد. در این کدها برای سرویس‌دهی به اجزای مختلف از جمله پورت PS/2 و پورت صدا از interrupt استفاده شده است و توابع ISR (Interrupt Service Routine) برای هر کدام در فایل جداگانه‌ای نوشته شده است. این کدهای C یکی از برنامه‌های آماده‌ی Altera Monitor Program است که در این آزمایش تغییرات مد نظر خود را در آن‌ها اعمال خواهیم کرد. از عملکرد صحیح کد اطمینان حاصل نمایید.

## ۲- نوشتن درایور برای راه‌اندازی ماوس با خروجی PS/2

یک درایور از مجموعه‌ای از توابع نرم‌افزاری تشکیل شده است که به برنامه‌های کامپیوتری اجازه می‌دهد بدون این که از جزئیات پیچیده و سطح پایین یک وسیله خارجی آگاهی داشته باشند با آن وسیله خارجی ارتباط داشته باشند. از مهم‌ترین وسایل خارجی مورد استفاده می‌توان به ماوس و کیبورد اشاره کرد. در این بخش هدف پیاده‌سازی یک درایور برای راه‌اندازی ماوس با رابط PS/2 است.

برای ارتباط با پورت PS/2 در DE2 Media Computer یک IP Core استفاده شده است. اطلاعات کامل در مورد این IP Core در فایل راهنمای آن قرار دارد. این هسته سخت‌افزاری، یک بافر FIFO ۲۵۶ بایتی دارد که اطلاعات را از PS/2 دریافت و ذخیره می‌کند. دسترسی به این بافر و نیز فعال کردن interruptها برای پورت PS/2 از طریق دو رجیستر memory-mapped انجام می‌شود. شکل ۴ این دو رجیستر را نشان می‌دهد.

Address	31	...	16	15	...	10	9	8	7	...	1	0		
0x10000100	RAVAIL			RVALID		Unused			Data				PS2_Data	
0x10000104							CE		RI				RE	PS2_Control

شکل ۴ رجیسترهای IP Core مورد استفاده برای پورت PS/2.

رجیستر PS2\_Data قابلیت خواندن و نوشتن دارد. هنگامی که بیت ۱۵ (RVALID) یک باشد با خواندن این رجیستر اولین داده موجود در بافر در فیلد Data و تعداد دیتای موجود در بافر در فیلد RAVAIL خوانده می‌شود (این تعداد شامل دیتای خوانده شده است). همچنین با این کار RAVAIL یک عدد کاهش یافته و Data خوانده شده نیز از بافر FIFO داخلی حذف می‌شود. با نوشتن در فیلد Data از رجیستر PS2\_Data می‌توان دستورات را به دستگاه PS/2 (که در اینجا ماوس است) انتقال داد.

رجیستر PS2\_Control برای فعال کردن interruptها به کار می‌رود. زمانی که RE=1 باشد، پورت PS/2 یک درخواست وقفه را وقتی که RAVAIL بزرگتر از صفر باشد، تولید می‌کند. RI=1 زمانی اتفاق می‌افتد که درخواست وقفه در حال تعلیق است و با خالی کردن FIFO می‌توان این بیت را صفر کرد. بیت CE نشانگر رخداد خطا در هنگام ارسال دستور به دستگاه PS/2 است.

دستوراتی که می‌توان به یک ماوس PS/2 ارسال کرد و جوابی که در ازای هر دستور دریافت می‌شود، در جدول ۱ آمده است. برای این که ماوس عملیات صحیحی انجام دهد، باید ابتدا ماوس را ریست کرد. پس از ارسال موفقیت آمیز دستور ریست (0xFF) انتظار می‌رود ماوس سه بایت متوالی 0xFAAA00 را بفرستد. بعد از دریافت پاسخ از ماوس می‌توان آن را با ارسال دستور 0xF4 فعال کرد و منتظر پاسخ 0xFA ماند.

جدول ۱ لیست دستورات و پاسخ‌های مورد انتظار از ماوس.

Command Description	Command Byte	Response
Reset the mouse to default mode	0xFF	Responds with a 0xFA message, followed by a 2-byte message 0xAA00 if successful. A byte 0xFC will be sent otherwise to indicate an error.
Enable Mouse to send position and button status messages	0xF4	Responds with a single 0xFA byte if successful.
Disable Mouse	0xF5	Responds with a single 0xFA byte if successful. Send the Enable mouse command to resume receiving messages about the user's interactions with the mouse.

در حالت پیش‌فرض وقتی یک ماوس فعال می‌شود (برای مثال هنگام حرکت آن) یک بسته (Packet) ۳ بایتی به پورت PS/2 می‌فرستد تا حالت خود را نشان دهد. فرمت بسته در جدول ۲ نشان داده شده است.

جدول ۲ اطلاعات ارسالی ماوس در قالب یک بسته ۳ بایتی.

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
byte 1	<i>y</i> overflow	<i>x</i> overflow	<i>y</i> sign	<i>x</i> sign	1	Middle btn	Right btn	Left btn
byte 2	Mouse <i>x</i> movement							
byte 3	Mouse <i>y</i> movement							

همانطور که مشاهده می‌شود تغییر در موقعیت ماوس با دو عدد ۹ بیتی مشخص می‌کند که یکی برای تغییرات در موقعیت‌های افقی و دیگری عمودی را مشخص می‌کند. اندازه حرکت در بایت‌های دوم و سوم و جهت حرکت در قالب بیت علامت (بیت‌های ۴ و ۵ از بایت اول) ارسال می‌شود. حرکت ماوس به سمت چپ با تغییرات افقی مثبت و حرکت ماوس به سمت راست با تغییرات افقی منفی بیان می‌شود. همچنین حرکت به سمت جلو و عقب نیز به ترتیب با تغییرات مثبت و منفی عمودی مشخص می‌شود. وضعیت سه دکمه ماوس با سه بیت پایین از بایت اول مشخص می‌شود که وقتی یک دکمه فشار داده شده باشد، بیت متناظر آن یک و در غیر این صورت صفر است. بیت شماره ۳ در بایت اول با یک شدن نشان می‌دهد که ۲ بایت بعدی متعلق به این بسته هستند.

برای پیاده‌سازی درایور ماوس PS/2 مراحل زیر را انجام دهید:

(۱) یک تابع C بنویسید که پورت PS/2 را ریست کند و ریست صحیح آن را تأیید کند. پس از ریست کردن این تابع باید ماوس را فعال کند و اگر نتواند آن را فعال کند مقدار 0 یا false را برگرداند. در نهایت این تابع interrupt ماوس را باید فعال کند (تابع initializeDriver).



(۲) یک تابع C بنویسید که بر اساس بسته دریافتی حالت دکمه را بر روی LEDR[2..0]، تغییرات در حرکت افقی (x) را بر روی HEX[7..4] و تغییرات در حرکت عمودی (y) را بر روی HEX[3..0] نمایش دهد. تغییرات لازم را در کد اولیه اعمال کنید (تابع getMouseChange).

(۳) در ادامه قابلیت درایور خود را به وسیله نگهداری مکان ماوس و محدود کردن مکانش به مرزهای مشخص افزایش می‌دهید. برای این کار شما باید ۵ متغیر تعریف کنید. دو متغیر برای نگهداری موقعیت افقی و عمودی، یکی برای نگهداری موقعیت دکمه ماوس و دو تا برای نگهداری ماکزیمم مختصات افقی و عمودی که ماوس می‌تواند در آن قرار بگیرد. توابع زیر را به کد اضافه کنید. تابع setMouseBounds(maxx, maxy) ماکزیمم مختصاتی که ماوس می‌تواند در آن قرار بگیرد را تعیین می‌کند. تابع getMouseState موقعیت فعلی ماوس بر روی صفحه و وضعیت ۳ دکمه را برمی‌گرداند. این دو تابع را پیاده‌سازی کنید. همچنین مقدار ماکزیمم افقی و عمودی را به ترتیب روی ۳۱۹ و ۲۳۹ ست کنید.

### ۳- طراحی Audio Player با نمایش گرافیکی و ایجاد Echo

در این بخش سیستم کلی پیاده‌سازی و تست خواهد شد. بدین منظور مراحل زیر را انجام دهید:

(۱) با مطالعه‌ی راهنمای DE2 Media Computer و نیز کدهای نمونه با نحوه کار با ورودی/خروجی آنالوگ آشنا شوید. تابع VGA\_box را مطالعه کنید تا با نحوه نمایش روی صفحه آشنا شوید. همچنین با نحوه کار با VGA Pixel Buffer و VGA Character Buffer آشنا شوید.

(۲) با استفاده از درایور ماوس خود، روی صفحه یک مکان‌نما (pointer) به صورت مربعی کوچک ایجاد کنید. در صورت استفاده از pointer به صورت فلش نمره امتیازی تعلق می‌گیرد. برای حرکت دادن ماوس در صفحه باید از قابلیت Back Buffer مربوط به Pixel Buffer DMA Controller استفاده نمایید و فهم عملکرد دقیق آن برای این کار ضروری است.

(۳) روی صفحه نمایش با استفاده از تابع VGA\_box مربع‌هایی به صورت دکمه برای ضبط، پخش و پخش همراه با echo ایجاد کنید. توجه نمایید که کد اصلی قابلیت ضبط و پخش صدا با استفاده از کلیدهای دوم و سوم را دارد. این امکان را از کد حذف نکنید.

(۴) یک تابع بنویسید که در صورت کلیک روی هر یک از این دکمه‌های عملیات مربوط به آن را انجام دهد.

(۵) به منظور echo کردن تابعی بنویسید که نمونه‌های  $0.25 \times [n-1000]$  و  $0.25 \times [n-5000]$  را با صدای خروجی جمع کند و سپس آن را پخش کند.

(۶) برنامه خود را روی برد تست کنید.

## بخش امتیازی

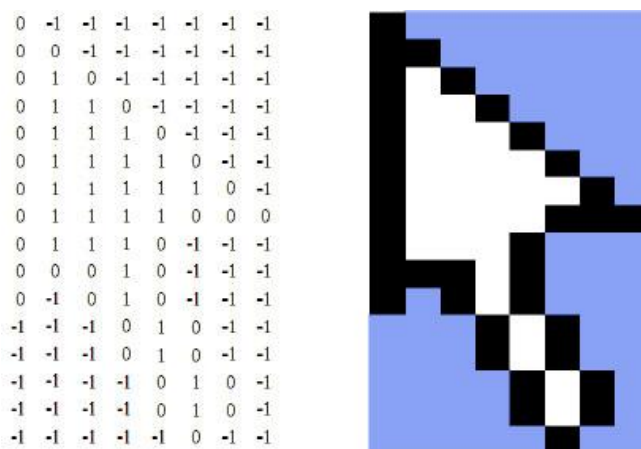
برای تعریف شکل ماوس یک ماتریس  $16$  در  $8$  ایجاد کنید. هر درایه ماتریس دارای یکی از این سه مقدار است:

0: نشان می‌دهد پیکسل متناظر اشاره‌گر ماوس مشکی است.

1: نشان می‌دهد پیکسل متناظر اشاره‌گر ماوس سفید است.

-1: نشان می‌دهد پیکسل متناظر اشاره‌گر ماوس شفاف است. این پیکسل‌ها هنگام حرکت ماوس رسم نمی‌شوند.

برای مثال شکل ۵ یک نمونه از اشاره‌گر ماوس و ماتریس متناظرش را نشان می‌دهد. رنگ آبی همان پیکسل‌های شفاف هستند.



شکل ۵ ماتریس و تصویر اشاره‌گر ماوس.

## نکات مهم:

(۱) در این آزمایش عملکرد صحیح سیستم اهمیت زیادی دارد. به عنوان مثال نباید تصویر پرش داشته باشد. کیفیت صدا باید مطلوب باشد. تشخیص مشکلات، تحلیل آن‌ها و حل مشکلات سیستم بخشی از این آزمایش است.

(۲) نیاز است هر آنچه را که در بخش ۱ خواسته شده است به دقت مطالعه نمایید. این کار بخشی از انجام آزمایش دوم است و امکان راهنمایی در این زمینه وجود ندارد. در مورد کد، خواندن کامنت‌های موجود

در کد می‌تواند کمک شایانی در فهم دقیق آن کند. برای انجام این آزمایش که کاملاً نرم‌افزاری است، لازم است کاربرد `interval_timer`، `VGA_pixel_buffer`، و همچنین تمامی `ISR`های موجود را کاملاً تحلیل نمایید.

(۳) به طور کلی ارتباط Nios با IP Core ها با خواندن و تغییر رجیسترهای آن‌ها (در آدرس مشخص خودشان) امکان پذیر است (همچنین ارتباط از طریق اینترپت و نیز DMA ها دیگر مجموعه ابزارهای ارتباطی بین Nios و سایر IP Core ها را تشکیل می‌دهند). درایورها شامل روال‌هایی سطح بالا از این ارتباطات هستند. به عنوان مثال در کد مربوط به بخش دوم ارتباط با `interval_timer` را (که در `bsp` درایور مناسبی برایش تعریف نشده است) با سایر IP Core ها مقایسه نمایید.

(۴) لزومی به تعریف توابع مطابق متن آزمایش وجود ندارد اما نتایج خواسته شده باید به دست آید. روند این آزمایش بدین صورت است: به هر طریقی که ممکن است، مکان ماوس را تعریف و به صفحه نمایش (طول ۰ تا ۳۱۹ و عرض ۰ تا ۲۳۹) محدود کنید و آن را روی `HEX Display` نشان دهید به طوری که با حرکت ماوس مکان آن نشان داده شود. سپس ماوس را به صورت یک مربع (استفاده از شکل فلش امتیازی است) در مکان مد نظر رسم نمایید (باید با حرکت ماوس شکل مکان قبلی آن بازگردد). دکمه های ضبط، پخش و پخش صدا همراه با اکو را ایجاد کنید و عملکردهای مربوطه را به آن‌ها اختصاص دهید (با کلیک روی آن‌ها عملکرد مربوطه اجرا شود). توجه نمایید که دو عملکرد ضبط و پخش صدا در کد اولیه پیاده‌سازی شده است.

(۵) کد `ISR` ها را محدود و کم کنید. سعی کنید ارتباط بین `ISR` با برنامه اصلی را با متغیرهای مشترک کنترل کنید و از انتقال برنامه های زمانبر به داخل `ISR` ها پرهیز کنید.

(۶) پیروی از قالب خاصی در گزارش مد نظر نیست، اما ترجیحاً می‌توانید از قالب ارائه شده برای تکالیف کامپیوتری استفاده نمایید.

(۷) آپلود کدهای C به همراه فایل گزارش ضروری است. لازم است کدهای بخش‌های ۱، ۲ و ۳ را در پوشه‌های جداگانه آپلود کنید (همه را در داخل یک فایل `zip`. قرار دهید).

موفق باشید

۹۷/۸/۱۰