

## مقدمه

دنیای دیجیتال امروز خدماتی را برای مردم فراهم کرده که زندگی را آسان‌تر و ایمن‌تر ساخته است. امکاناتی که سیستم‌های دیجیتال در اختیار کاربران قرار می‌دهد علی‌رغم تعدد و تنوع، در محصولی بسیار کوچک مانند ساعت گوگل قرار می‌گیرد. یکی از این امکانات، سیستم مراقبت از سلامت است.

## شرح کلی سامانه

پروژه‌ی پایانی درس مدارهای منطقی طراحی یک سامانه‌ی مراقبت از سلامت است که از کاربر ویژگی‌های فردی و حیاتی متفاوتی را دریافت کرده و با تحلیل و بررسی آن‌ها، هشدارهای متفاوتی را به کاربر اعلام می‌کند. هدف از این پروژه آشنایی دانشجویان با طراحی یک سامانه‌ی نهفته بی‌درنگ<sup>۱</sup> است. دانشجویان باید هسته‌ی پردازشی<sup>۲</sup> اصلی سامانه را طراحی نمایند.

این سامانه با استفاده از اطلاعات ورودی، معیارهای سلامتی شامل فشار خون، دمای بدن، رخداد شوک عصبی یا شوک قلبی را اندازه‌گیری می‌کند و اطلاعات مربوط به هر بخش را در خروجی نشان می‌دهد. علاوه بر این، در صورتی که شرایط بحرانی (مانند فشار خون بالا، تب، شوک عصبی و قلبی) رخ دهد، سامانه بلافاصله شخص را از این موضوع آگاه می‌سازد. در این سامانه اطلاعات شخصی فرد شامل سن، نمایه‌ی توده‌ی بدن<sup>۳</sup>، غلظت خون و نوع گروه خونی قابل تنظیم است که بر نحوه‌ی محاسبه‌ی علائم حیاتی تأثیرگذار است.

## فاز اول: طراحی بخش‌های ترکیبی

در فاز اول پروژه، باید از مدارهایی که در "آزمایشگاه مدارهای منطقی" طراحی و در سامانه درس بارگذاری کرده‌اید، استفاده کرده و بخشی از پروژه نهایی درس را پیاده‌سازی کنید.

**توجه:** در هنگام طراحی و پیاده‌سازی، نام متغیرها را همانند آنچه که در شکل‌ها و کدها آمده است تعریف کنید. در غیر این صورت، مدار پیاده‌سازی شده‌ی شما با فایل محیط آزمون (فایل شبیه‌سازی) به درستی کار نخواهد کرد و پیاده‌سازی بر روی بورد FPGA امکان‌پذیر نخواهد بود.

### لطفاً هنگام پیاده‌سازی نکات زیر را در نظر بگیرید:

- برای هر یک از قطعه‌کدها، یک فایل با پسوند v قرار داده شده است. یک پروژه در محیط Xilinx ISE Suite داده شده است. برای مدار نهایی فاز اول فایل محیط آزمون جهت شبیه‌سازی طراحی شده است. می‌توانید جهت اطمینان از درستی طراحی از آن استفاده نمایید. یک فایل طراحی جهت بررسی درستی مدار کلی بر روی بورد طراحی شده است که در روز تحویل پروژه به صورت جداگانه در اختیار شما قرار می‌گیرد.
- برای هر یک از ماژول‌ها باید ابتدا توابع و شماتیک مدار را طراحی نمایید. می‌توانید بر روی کاغذ بنویسید یا از ابزاری مانند Microsoft Visio استفاده نمایید. هنگام تحویل پروژه باید توابع و شماتیک طرح را نیز ارائه دهید.
- پروژه به صورت گروهی است (اعضای آزمایشگاه مدارهای منطقی) ولی تسلط هر یک از اعضا به کل پروژه الزامی است. نمره بخش طراحی توابع و شماتیک طرح به صورت گروهی اعمال خواهد شد ولی بخش عمده‌ای از نمره فردی است.
- هرگونه رونوشت و همکاری با سایر گروه‌ها تقلب محسوب شده و موجب تبعات سنگین خواهد شد. پروژه‌ی تمامی ۱۲ گروه آزمایشگاه با هم مقایسه خواهد شد.

<sup>1</sup> Real-time embedded system

<sup>2</sup> Processing Core

<sup>3</sup> Body-Mass-Index (BMI)

- تحویل پروژه به صورت حضوری است و همه اعضای گروه موظف‌اند حضور داشته باشند و به صورت مستقل به پرسش‌های شفاهی پاسخ دهند. پروژه را باید بر روی بورد FPGA پیاده‌سازی کرده و نتیجه بگیرید.
- زمان تحویل این فاز از پروژه، جلسه‌ی بعد از توضیح صورت پروژه توسط مدرس آزمایش خواهد بود. به دلیل فشرده بودن زمان آزمایشگاه، امکان تمدید پروژه وجود ندارد.
- ۳ ساعت قبل از شروع جلسه‌ی تحویل فاز اول پروژه، حتماً فایل‌های طراحی شده خود را در سامانه‌ی کوئرا بارگذاری کنید. همه‌ی اعضای گروه فایل‌های فاز اول پروژه را بارگذاری نمایند.

نمره (درصد)	اجزای مدار ترکیبی سامانه مراقبت از سلامت	
۱۰	پیاده‌سازی مدار افزونگی بیت توازن	بخش ۱ (واحد فشار خون)
۱۰	پیاده‌سازی تشخیص‌دهنده‌ی غیرطبیعی بودن فشار خون	
۰	پیاده‌سازی مدار نهایی بخش ۱	
۱۰	پیاده‌سازی مدار تفکیک‌کننده‌ی گروه خونی	بخش ۲ (واحد غلظت خون)
۱۰	پیاده‌سازی مدار تشخیص‌دهنده‌ی طبیعی بودن غلظت PH خون	
۰	پیاده‌سازی مدار نهایی	
۸	پیاده‌سازی مدار مقایسه‌کننده ۸ بیتی	بخش ۳ (واحد تشخیص سقوط)
۰	پیاده‌سازی مدار نهایی	
۱۲	پیاده‌سازی مدار محاسبه‌کننده‌ی دمای بدن	بخش ۴ (واحد محاسبه دمای بدن)
۰	پیاده‌سازی مدار تشخیص غیرطبیعی بودن دمای بدن	
۰	پیاده‌سازی مدار کلی	
۸	پیاده‌سازی مدار قدرمطلق گیر	بخش ۵ (فاز اختیاری: تشخیص قند خون)
۸	پیاده‌سازی تشخیص‌دهنده‌ی قند خون	
۴	پیاده‌سازی مدار نهایی فاز اختیاری	
۱۰	درستی کارکرد مدار کلی فاز اول (اتصال ماژول‌های طراحی شده + خروجی مناسب مدار کلی)	بخش ۶ (مدار کلی فاز اول)
۱۰	گزارش توابع و مدارهای طراحی شده	گزارش
۲۰	پاسخ به پرسش‌های شفاهی، تسلط به ابزار و کد نوشته شده	تسلط

## بخش ۱: مدار فشار خون

### افزونگی بیت توازن

حسگر فشار خون به دلیل وجود نویز به داده‌ی اندازه‌گیری شده یک بیت افزونگی توازن زوج اضافه می‌کند. لذا در هنگام دریافت باید درستی داده بررسی شود. اگر داده‌ی ورودی به سامانه معتبر باشد، باید بررسی شود که ورودی از مقدار آستانه تجاوز نکرده باشد. در صورتی که داده معتبر نباشد، خروجی مدار که غیرطبیعی بودن فشار خون را تشخیص می‌دهد نادیده گرفته خواهد شد. قالب داده‌ی ورودی سامانه در شکل ۱-۱ آمده است.

5	4	3	2	1	0
Even Parity			Pressure Data		

شکل ۱-۱ قالب داده‌ی ورودی مدار فشار خون

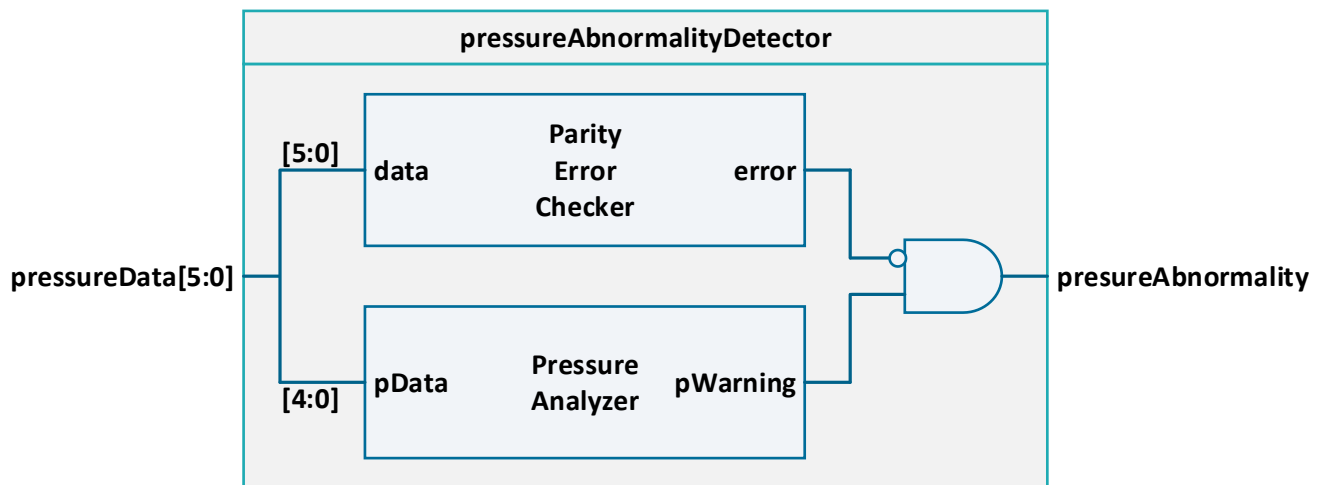
- ✎ **الف)** با استفاده از مدار طراحی شده در «آزمایش شماره ۴ آزمایشگاه مدارهای منطقی»، مدار افزونگی مناسبی طراحی نمائید که در صورتی که داده‌ی ۶ بیتی فوق دارای خطا باشد، آن را تشخیص دهد (خروجی = ۱).
- ✎ **ب)** مدار طراحی شده در قسمت «الف» را با استفاده از زبان وریلاگ توصیف نمایید. قطعه‌کد ۱-۱ ورودی‌ها و خروجی‌ها را مشخص کرده است.

### تشخیص‌دهنده‌ی غیرطبیعی بودن فشار خون

- مدار تشخیص‌دهنده‌ی غیرطبیعی بودن فشار خون، دارای یک ورودی ۵ بیتی بدون علامت (مقدار فشار اندازه‌گیری شده توسط حسگر) است. برای یک فرد مقدار فشار خون در بازه‌ی ۸ الی ۲۲ حالت طبیعی و در غیر این صورت، فشار غیرطبیعی است.
- ✎ **ج)** تابع مجموعه مینترم‌های مداری را مشخص کنید که ورودی آن یک عدد ۵ بیتی بدون علامت است و خروجی آن مشخص می‌کند که داده در بازه‌ی ۸ الی ۲۲ (شامل ۸ و ۲۲) قرار دارد (خروجی = ۰) یا خیر (خروجی = ۱).
- ✎ **د)** تابع قسمت «ج» را با استفاده از جدول کارنو ساده کرده و آن را به صورت مجموع حاصل ضرب‌ها (Sum of Products) بنویسید.
- ✎ **ه)** مدار طراحی شده در قسمت «د» را با استفاده از زبان وریلاگ توصیف نمایید. قطعه‌کد ۱-۲ ورودی‌ها و خروجی‌ها را مشخص کرده است.

### مدار نهایی بخش ۱

- سامانه‌ی مراقبت از سلامت، در صورتی هشدار را در نظر می‌گیرد که داده‌ی ورودی مسئله معتبر باشد. به عبارت دیگر، اگر به دلیل عدم توازن مقدار دریافت شده، اشتباه باشد، نباید هشدار داده شود. شکل ۱ مدار فشار خون (بخش ۱) را مشخص کرده است.
- ✎ **و)** با استفاده از قطعه‌کدهای طراحی شده در دو بخش قبلی، مدار شکل ۱-۱ را با استفاده از زبان وریلاگ توصیف نمایید. قطعه‌کد ۱-۳ ورودی و خروجی‌ها را مشخص کرده است.



شکل ۱-۱ مدار تشخیص‌دهنده‌ی غیرطبیعی بودن فشار خون

قطعه‌کد ۱-۱: تشخیص خطای داده‌ی ارسالی حسگر فشار خون

```
module parityErrorChecker (
    data,
    error);
input [5:0] data;
output error;
    // write your code here, please.
endmodule
```

قطعه‌کد ۲-۱: تشخیص‌دهنده‌ی غیرطبیعی بودن فشار خون

```
module pressureAnalyzer (
    pData,
    pWarning);
input [4:0] pData;
output pDarning;
    // write your code here, please.
endmodule
```

قطعه‌کد ۳-۱: مدار کلی تشخیص‌دهنده‌ی غیرطبیعی بودن فشار خون

```
module pressureAbnormalityDetector (
    pressureData,
    presureAbnormality);
input [5:0] pressureData;
output presureAbnormality;
    // write your code here, please.
endmodule
```

## بخش ۲: مدار غلظت خون

### مدار تفکیک‌کننده‌ی گروه خونی

در سامانه‌ی مراقبت از سلامت، نوع گروه خونی بر طبیعی بودن غلظت PH خون تأثیرگذار است. لذا ابتدا باید یک مدار تفکیک‌کننده‌ی گروه خونی طراحی شود. در جدول ۱-۲ کد مربوط به هر نوع گروه خونی آمده است.

جدول ۱-۲ کد مربوط به نوع گروه خونی

کد مربوطه	گروه خونی
000	$O^-$
001	$O^+$
010	$A^-$
011	$A^+$
100	$B^-$
101	$B^+$
110	$AB^-$
111	$AB^+$

الف) با توجه به جدول ۱-۲ تابعی طراحی نمایید که اگر گروه خونی فرد از نوع O یا B+ باشد خروجی آن برابر یک در غیر این صورت صفر باشد.

ب) تابع قسمت «الف» را با استفاده از یک مالتی‌پلکسر ۴ به ۱ و بدون استفاده از هیچ گیت دیگری پیاده‌سازی کنید.

ج) مدار طراحی‌شده در قسمت «ب» را با استفاده از زبان ورپلاگ توصیف نمایید. قطعه‌کد ۱-۲ ورودی و خروجی‌های کد را مشخص کرده است.

### مدار تشخیص‌دهنده‌ی طبیعی بودن غلظت PH خون

برای همه‌ی گروه‌های خونی مقدار غلظت ۷ و ۸ حالت طبیعی است. برای گروه خونی O و B+ علاوه بر این مقادیر، مقادیر ۶ و ۹ نیز حالت طبیعی است. بنابراین سامانه‌ی تشخیص‌دهنده‌ی طبیعی بودن غلظت PH باید با توجه به مقدار و نوع گروه خونی تصمیم بگیرد.

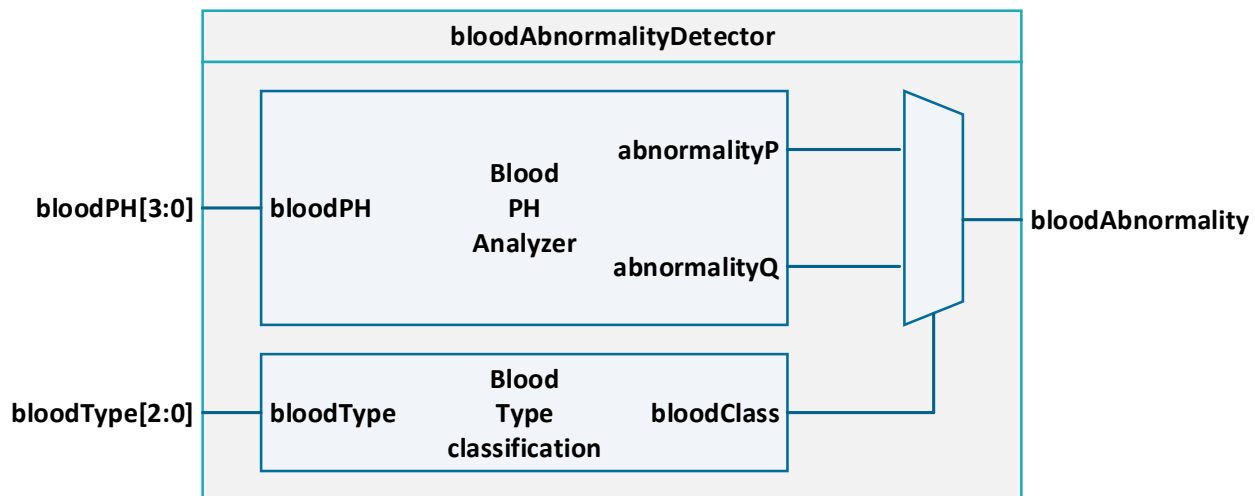
د) توابع مربوط به abnormalityP و abnormalityQ را طراحی نمایید.

- منظور از abnormalityP غیرطبیعی بودن غلظت PH خون برای گروه خونی غیر O و B+ است.

- منظور از abnormalityQ غیرطبیعی بودن غلظت PH خون برای گروه خونی O و B+ است.

و) مدار طراحی‌شده در قسمت «د» را با استفاده از زبان ورپلاگ توصیف نمایید. قطعه‌کد ۲-۲ ورودی و خروجی‌های کد را مشخص کرده است.

ه) با استفاده از قطعه‌کدهای طراحی شده قبلی، مدار شکل ۱-۲ را با استفاده از زبان ورپلاگ توصیف نمایید. قطعه‌کد ۳-۲ ورودی‌ها و خروجی کد را مشخص کرده است.



شکل ۱-۲ نمای کلی فاز دوم پروژه (تشخیص غیرطبیعی بودن غلظت خون)

قطعه کد ۱-۲: تفکیک‌کننده‌ی گروه خونی

```
module bloodTypeClassification(
    bloodType,
    bloodClass);
input [2:0] bloodType;
output bloodClass;
    // write your code here, please.
endmodule
```

قطعه کد ۲-۲: تشخیص غیرطبیعی بودن غلظت PH خون

```
module bloodPHAnalyzer(
    bloodPH,
    abnormalityP,
    abnormalityQ);
input [3:0] bloodPH;
output abnormalityP;
output abnormalityQ;
    // write your code here, please.
endmodule
```

قطعه کد ۳-۲: مدار کلی تشخیص غیرطبیعی بودن غلظت PH خون

```
module bloodAbnormalityDetector(
    bloodPH,
    bloodType,
    bloodAbnormality);
input [3:0] bloodPH;
input [2:0] bloodType;
output bloodAbnormality;
    // write your code here, please.
endmodule
```

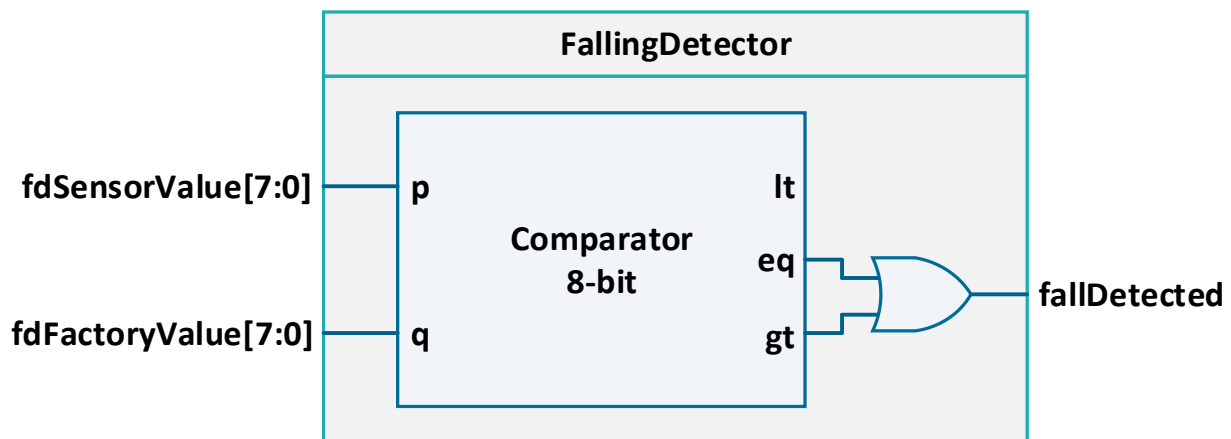
### بخش ۳: مدار تشخیص دهنده‌ی سقوط

#### مدار مقایسه کننده

در سامانه‌ی مراقبت از سلامت، حسگری<sup>۴</sup> وجود دارد که برای تشخیص افتادن، سقوط و از دست دادن تعادل فرد، مورد استفاده قرار می‌گیرد. خروجی این حسگر یک عدد ۸ بیتی بدون علامت است.

سامانه‌ی کنترل کننده باید این مقدار را دریافت کرده و با یک عدد ۸ بیتی بدون علامت که توسط کارخانه‌ی سازنده‌ی حسگر تعیین می‌شود مقایسه کند. در صورتی که خروجی حسگر از مقدار مشخص شده توسط کارخانه بزرگتر یا مساوی باشد، سامانه باید تشخیص دهد که فرد تعادل خود را از دست داده است و هشدار مورد نظر را تولید نماید.

✎ **الف)** مدار شکل ۱-۳ را با استفاده از زبان وریلاگ توصیف نمایید. قطعه‌کد ۱-۳ ورودی و خروجی‌های کد را مشخص کرده است. (جهت پیاده‌سازی این بخش از مقایسه کننده‌ی ۸ بیتی که در آزمایش شماره ۶ آزمایشگاه مدارهای منطقی طراحی کرده‌اید استفاده کنید).



شکل ۱-۳ مدار تشخیص دهنده‌ی سقوط و عدم تعادل

قطعه‌کد ۱-۳: تشخیص دهنده‌ی سقوط

```
module fallingDetector(
    fdSensorValue,
    fdFactoryValue,
    fallDetected);
input [7:0] fdSensorValue;
input [7:0] fdFactoryValue;
output fallDetected;

    // write your code here, please.

endmodule
```

<sup>4</sup> Sensor

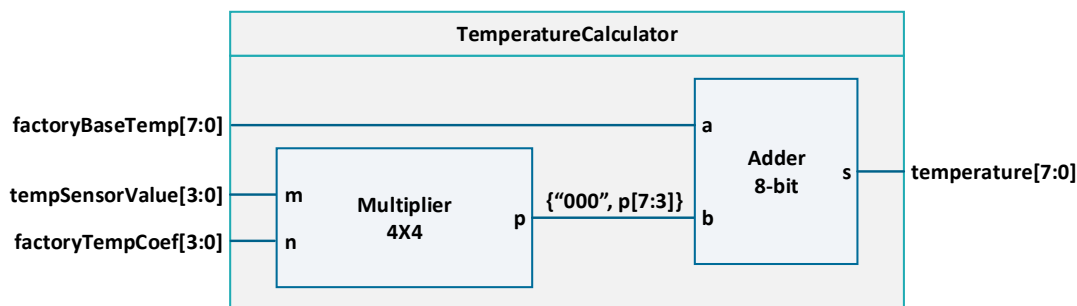
## بخش ۴: مدار محاسبه‌کننده‌ی دمای بدن

### مدار محاسبه‌کننده‌ی دمای بدن

فرض کنید در سامانه‌ی مراقبت از سلامت یک دماسنج قرار داده شده است. ارتباط با دماسنج از طریق مبدل آنالوگ به دیجیتال انجام می‌گیرد. در این سامانه ابتدا دمای بدن به وسیله‌ی ماژول دماسنج به مقدار ولتاژ مناسب تبدیل می‌شود. سپس ولتاژ که یک کمیت پیوسته و آنالوگ است، با استفاده مبدل آنالوگ به دیجیتال به داده‌ی دیجیتال تبدیل می‌شود. داده‌ی به دست آمده یک داده‌ی خام بوده و لزوماً برابر با مقدار دمای بدن نیست و باید به داده‌ی مناسبی تبدیل شود. از رابطه‌ی ۱ جهت تبدیل داده‌ی خام (خروجی مبدل آنالوگ به دیجیتال) به داده‌ی مورد انتظار (دمای بدن) استفاده می‌شود.

$$\text{temperature} = \text{factoryBaseTemp} + \left\lfloor \frac{(\text{factoryTempCoef} \times \text{tempSensorValue})}{8} \right\rfloor \quad \text{رابطه ۱}$$

- خروجی  $\text{temperature}$  دمای بدن فرد است.
  - ورودی  $\text{tempSensorValue}$  داده‌ی خام دریافتی از حسگر است.
  - ورودی  $\text{factoryTempCoef}$  ضریب محیطی است که توسط کارخانه تنظیم می‌شود.
- شکل ۴-۱ نحوه‌ی پیاده‌سازی این عملیات را نشان می‌دهد.

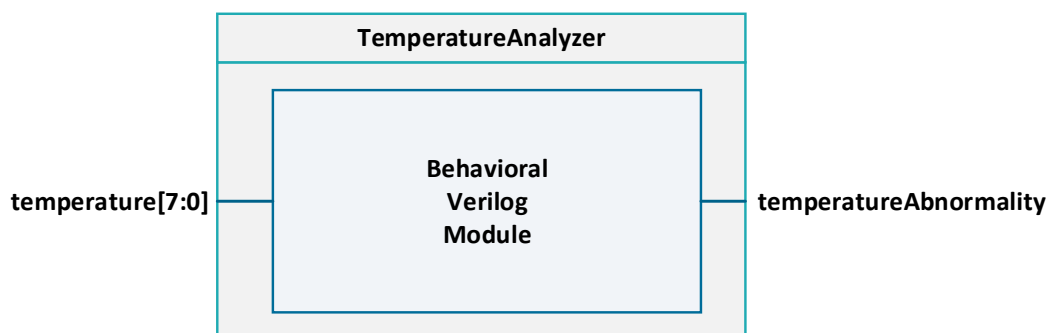


شکل ۴-۱ مدار محاسبه‌کننده‌ی رابطه‌ی ۱

**الف) مدار شکل ۴-۱ را با استفاده از زبان وریلاگ توصیف نمایید.** قطعه‌کد ۴-۱ ورودی‌ها و خروجی‌ها را مشخص کرده است. **توجه:** جهت پیاده‌سازی، از مدارهای طراحی شده در آزمایش‌های ۷ و ۸ آزمایشگاه مدارهای منطقی استفاده نمایید. استفاده از عملگرهای  $\times$  و  $+$  در زبان وریلاگ مجاز نیست.

### مدار تشخیص غیرطبیعی بودن دمای بدن

مدار شکل ۴-۲ جهت تشخیص تب یا دمای غیرطبیعی بدن به کار می‌رود. توصیف وریلاگ این مدار در اختیار دانشجویان قرار داده شده است. خروجی این مدار زمانی که دمای بدن بین ۳۵ و ۳۹ باشد (شامل آن‌ها) برابر با صفر خواهد بود. در غیر این صورت، برابر با یک می‌شود که معادل با غیرطبیعی بودن دمای بدن است. توصیف وریلاگ این ماژول (Temperature Abnormality Detector) در قطعه‌کد ۴-۲ آمده است.

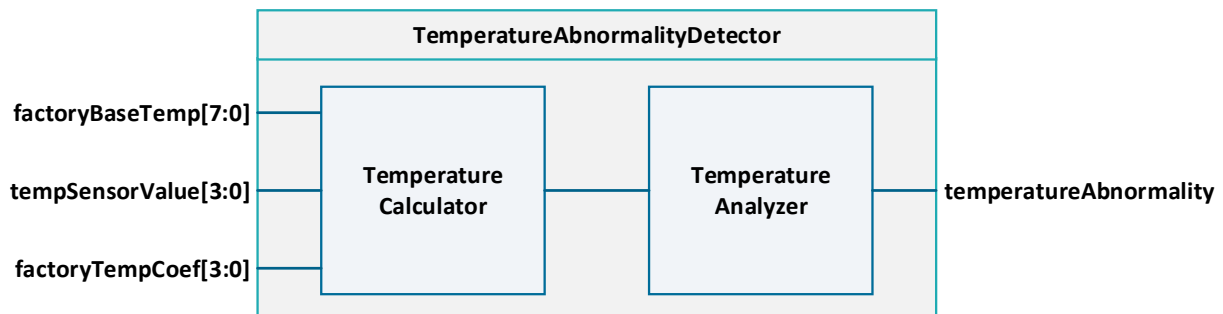




شکل ۲-۴ مدار تشخیص‌دهنده‌ی غیرطبیعی بودن دمای بدن

#### مدار کلی بخش ۴

ب) با استفاده از مدارهای طراحی قسمت «الف» و قطعه‌کد ۲-۴، مدار کلی تشخیص غیرطبیعی بودن دمای بدن (شکل ۳-۴) را با استفاده از زبان وریلاگ توصیف نمایید. قطعه‌کد ۳-۴ ورودی و خروجی کد را مشخص کرده است.



شکل ۳-۴ مدار کلی محاسبه‌کننده‌ی غیرطبیعی بودن دمای بدن

قطعه‌کد ۴-۱: مدار محاسبه‌کننده‌ی دمای بدن

```

module temperatureCalculator(
    factoryBaseTemp,
    factoryTempCoef,
    tempSensorValue,
    temperature);
input [7:0] factoryBaseTemp;
input [3:0] factoryTempCoef;
input [3:0] tempSensorValue;
output [7:0] temperature;
// write your code here, please.
endmodule
  
```

قطعه‌کد ۲-۴: مدار تشخیص غیرطبیعی بودن دمای بدن

```

module temperatureAnalyzer(
    temperature,
    temperatureAbnormality);
input [7:0] temperature;
output temperatureAbnormality;

assign temperatureAbnormality = (temperature > 39) || (temperature < 35);
endmodule
  
```

قطعه‌کد ۳-۴: مدار محاسبه‌کننده‌ی دمای بدن

```

module temperatureAbnormalityDetector(
    factoryBaseTemp,
    factoryTempCoef,
    tempSensorValue,
    temperatureAbnormality);
input [7:0] factoryBaseTemp;
input [3:0] factoryTempCoef;
input [3:0] tempSensorValue;
  
```

---

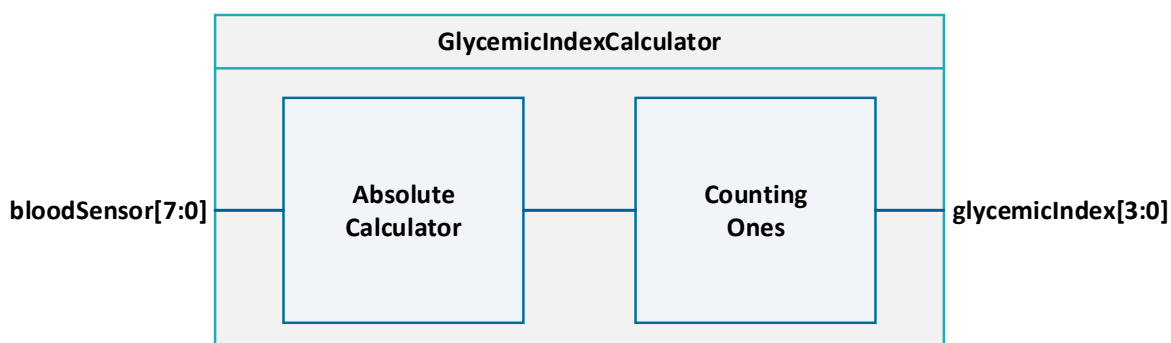
```
output temperatureAbnormality;  
    // write your code here, please.  
endmodule
```

---

## بخش ۵: مدار تشخیص قند خون (اختیاری فاز اول)

سامانه‌ی مراقبت از سلامت دارای حسگری است که مقدار قند خون را اندازه‌گیری می‌کند. خروجی حسگر یک عدد ۸ بیتی است. مدار محاسبه‌کننده‌ی میزان قند خون از دو بخش تشکیل شده است. ابتدا قدر مطلق داده‌ی ۸ بیتی گرفته می‌شود. سپس تعداد یک‌های بردار شمرده می‌شود. تعداد بیت‌های یک، مقدار قند خون خواهد بود.

**الف)** مدار شکل ۱-۴ یک حالت انتزاعی از طرح را نشان می‌دهد. پس از طراحی توابع مورد نیاز، آن را با استفاده از زبان ورپلاگ توصیف نمایید. قطعه‌کد ۱-۵ ورودی و خروجی کد را مشخص کرده است. در طراحی خود می‌توانید از هر بلاک منطقی مانند گیت‌های پایه، جمع‌کننده، تفریق‌کننده، مقایسه‌کننده، دیکدر، انکدر و مالتی‌پلکسر استفاده نمایید. توجه داشته باشید که توصیف شما باید به صورت ساختاری (با ایجاد بلاک‌های منطقی و اتصال پورت‌های آن‌ها) باشد. همچنین نباید از توصیف رفتاری استفاده نمایید.



شکل ۱-۵ مدار محاسبه‌کننده‌ی قند خون

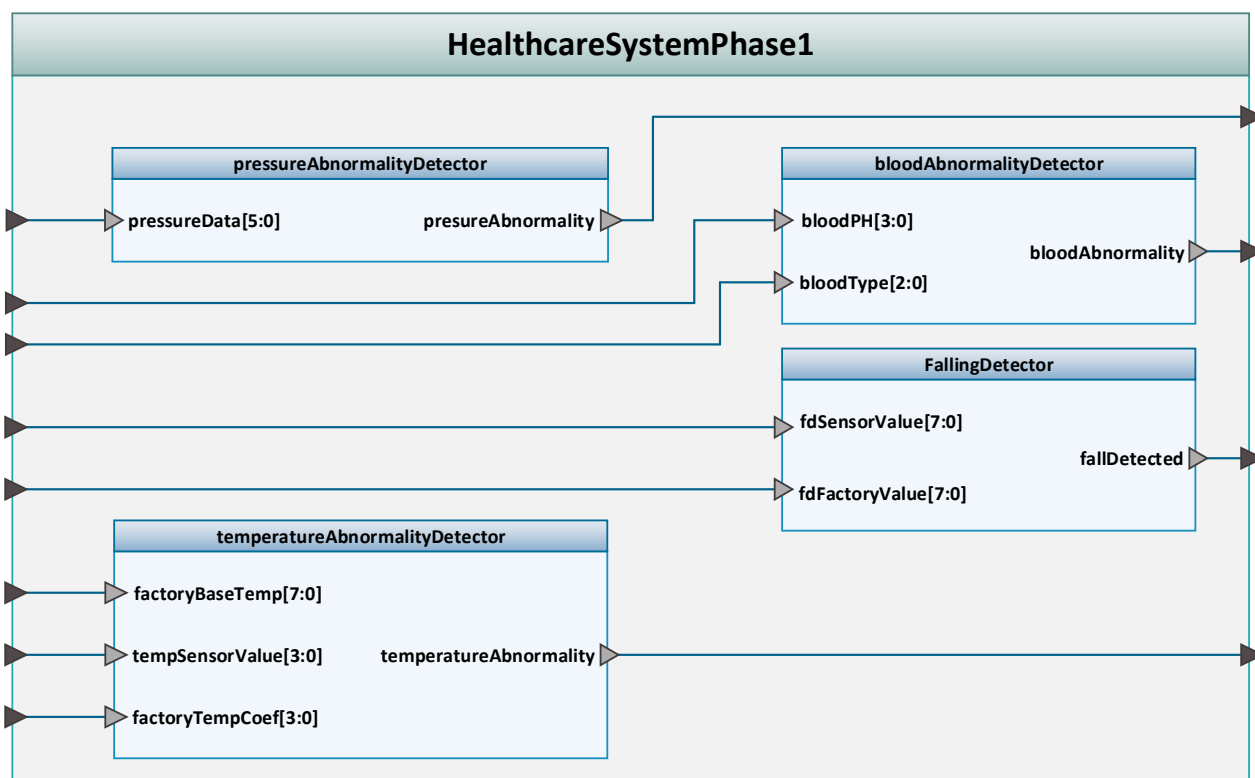
قطعه‌کد ۱-۵: مدار تشخیص‌دهنده‌ی قند خون

```
module glycemicIndexCalculator(
    bloodSensor,
    glycemicIndex);
input [7:0] bloodSensor;
output [3:0] glycemicIndex;
    // write your code here, please.
endmodule
```

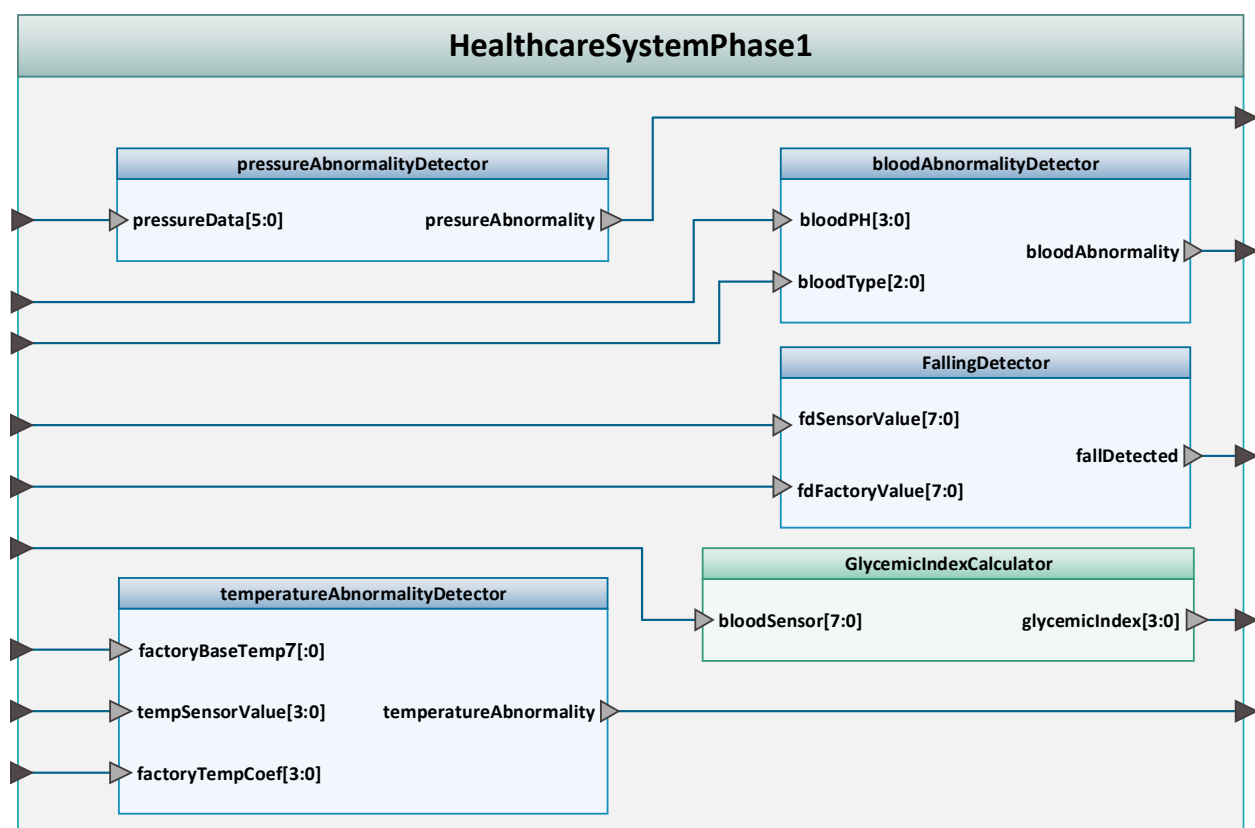
## بخش ۶: ترکیب مدارهای فاز اول پروژه

در این بخش مدارهای طراحی شده در بخش‌های قبلی را به هم متصل خواهید کرد و بخش بزرگی از پروژه‌ی نهایی درس (سامانه‌ی مراقبت از سلامت) را طراحی خواهید کرد. شکل ۱-۶ مدار کلی فاز اول پروژه را نشان می‌دهد (در شکل ۲-۶ نیز مدار کلی برای حالتی که بخش اختیاری نیز طراحی شده باشد آمده است). در قطعه‌کد ۱-۶ ورودی‌ها و خروجی‌های مدار آمده است (قطعه‌کد ۲-۶ برای حالتی است که بخش اختیاری نیز طراحی شده باشد).

توجه داشته باشید که نام ورودی‌ها و خروجی‌های مدار اصلی هم‌نام با نام زیر بخش‌ها است.



شکل ۶-۱ مدار کلی فاز اول سامانهی مراقبت از سلامت



شکل ۶-۲ مدار کلی فاز اول سامانهی مراقبت از سلامت شامل بخش اختیاری

قطعه‌کد ۶-۱: مدار کلی فاز اول سامانه‌ی مراقبت از سلامت

```
module HealthcareSystemFirstPhase (  
    pressureData,  
    bloodPH,  
    bloodType,  
    fdSensorValue,  
    fdFactoryValue,  
    factoryBaseTemp,  
    tempSensorValue,  
    factoryTempCoef,  
    presureAbnormality,  
    bloodAbnormality,  
    fallDetected,  
    temperatureAbnormality);  
  
input [5:0] pressureData;  
input [3:0] bloodPH;  
input [2:0] bloodType;  
input [7:0] fdSensorValue;  
input [7:0] fdFactoryValue;  
input [7:0] factoryBaseTemp;  
input [3:0] factoryTempCoef;  
input [3:0] tempSensorValue;  
  
output presureAbnormality;  
output bloodAbnormality;  
output fallDetected;  
output temperatureAbnormality;  
  
    // write your code here, please.  
  
endmodule
```

قطعه‌کد ۶-۲: مدار کلی فاز اول سامانه‌ی مراقبت از سلامت شامل بخش اختیاری

```
module HealthcareSystemPhase1 (  
    pressureData,  
    bloodPH,  
    bloodType,  
    fdSensorValue,  
    fdFactoryValue,  
    bloodSensor,  
    factoryBaseTemp,  
    tempSensorValue,  
    factoryTempCoef,  
    presureAbnormality,  
    bloodAbnormality,  
    fallDetected,  
    glycemcIndex,  
    temperatureAbnormality);  
  
input [5:0] pressureData;  
input [3:0] bloodPH;  
input [2:0] bloodType;  
input [7:0] fdSensorValue;  
input [7:0] fdFactoryValue;  
input [7:0] bloodSensor;  
input [7:0] factoryBaseTemp;  
input [3:0] factoryTempCoef;  
input [3:0] tempSensorValue;  
  
output presureAbnormality;  
output bloodAbnormality;  
output fallDetected;  
output [3:0] glycemcIndex;  
output temperatureAbnormality;  
  
    // write your code here, please.  
  
endmodule
```