Simulation of Driver Fatigue Monitoring via Blink Rate Detection, using 65nm CMOS Technology.

N. Yassine, S. Barker, K. Hayatleh, B. Choubey, R. Nagulapalli

This paper proposes a system to detect and measure blink rate to determine fatigue levels. The method involved analysing specific frames to determine that a blink occurred, and then monitoring the time between successive blinks. The program was simulated in python using a Raspberry Pi Zero and a standard USB camera. For the blink rate detection block, a gate level schematic was implemented in Cadence software using 65nm CMOS technology. The design was based around an asynchronous 6-bit based edge counter which was designed using D-flip-flops. The simulation calculated the average blink rate and compared this to the most recent blink rate. The outcome would determine if an alarm signal should be sent to the alarm. The system consumed 130uA from a 1.2V power supply.

Key Words: CMOS; Driver Fatigue; Blink Rate Detection; Low Power

Introduction

This paper proposes a design for a driver monitoring and warning system using the drivers' blink rate to determine their fatigue levels. Each year driver fatigue contributes to thousands of accidents, which are likely to lead to death or serious injury [1]. The resulting collisions can be due to the driver not reacting fast enough, because of slow reactions from reduced driver awareness caused by increasing levels of fatigue [1]. A survey showed that in 2012, 29% of drivers admitted having almost fallen asleep at the wheel [2]. The fatigued driver is not only putting themselves at risk but also other drivers, passengers and pedestrians. Developing a system that warns a driver of increased fatigue levels will allow them to take appropriate action, thereby reducing the risk of a collision and injury to themselves and others. Research has shown [2] that performance levels when under the influence of alcohol or sleep deprivation showed similar degradations. Hence, measures for providing warning of driver fatigue could have a significant impact in reducing the likelihood of accidents. The fatigue monitoring system outlined here has the potential to be used in any situation where fatigue causes an increased risk to safety. Security personnel working on x-ray machines at an airport, and airline pilots are examples of others that could benefit from this system. The aviation industry has noted that some notable airline accidents caused by pilot error, resulted from a direct influence of pilot fatigue [2], thereby demonstrating the safety benefits that this system would bring.

There are awareness-warning devices available today that will warn drivers when they get tired. When asking a pilot to describe their awareness level, they may not feel tired, but testing their reaction times, may show otherwise [2]. However, when confronting the pilot with a test, they can become more alert in preparation for it, which would provide inaccurate results [2]. Thus, BALPA (British Air Line Pilots' Association) uses an eyelid monitoring system using wearable technology from Optalert [2]. Optalert developed a wearable technology system to track the driver's eye and eyelid movement to detect fatigue levels before the driver themselves become aware of it [3]. The system involves alerting the driver when they get tired and stores large amounts of user data for ongoing research. The Optalert system uses infrared light to detect eyelid movement. The driver wears Optalert glasses that have infrared emitters and receivers mounted in the frame. The glasses also have a processor built in for on board data processing, which then transmits the data to the Optalert vehicle system [4]. The vehicle system monitors the incoming data and provides visual and audible alerts informing the driver they are starting to get tired [3]. This early detection system did have a major challenge during the company's' research and development of the system. The issue mentioned, involved the amount of raw data, on eye movement, that needed to be stored. The amount of eye movement data, as well as warnings and events, such as turning off the vehicle, amounted to around 2MB per user, per hour [3]. The issue with this system is it requires the user to wear glasses. The system presented here has the benefit that it is built into the car, hence no need for special glasses, which would have impaired the driver's ability to wear their own glasses.

An important consideration to consider is measuring the normal blink rate of the driver at the beginning of the journey. In a normal rest condition, a human's average blink rate is around 17 blinks per minute [5]. Depending on a specific activity, the blink rate can go as high as 26 blinks per minute, to as low as 4-5 blinks per minute [5]. While driving the blink rate can be lower than resting conditions, at around 4-5 blinks per minute [5]. Therefore it is vital to measure the blink rate at the beginning of the journey and use it as the 'new normal' rate.

Having looked at current existing technology, the proposed system has a number of requirements. These include affordability and ease of use, as well as provide accurate monitoring of fatigue levels. The warning system needs to be activated as early as possible to minimize risk of accident occurring from the driver feeling tired. For a broader range of use, the system should work in low light levels, so detection can work during night, which is when the driver is most likely going to feel tired.

Eye Detection: A key feature of the proposed system is eye detection. Various detection methods were considered, including the method proposed by Marc Lalonde *et al.* [6]. This involved simple motion detection, based on looking at the difference between frames inside the tracked regions of the eye [6]. The approach taken involved using Haar-cascade classifiers [7], which was initially proposed by Paul Viola and Michael Jones in 2001 [8].

Viola and Jones [8] proposal was based on the use of machine learning for visual object detection capable of highly successful image detection at high rates [8]. The detection system they proposed had three key features. The first is the integral image which provided high computation rates for the features used by their detector [8]. The second was their learning algorithm based on AdaBoost [9]. This effectively improved their learning algorithm by selecting specific features from a large data pool and providing efficient classifiers [8]. The third was combining the classifiers into a cascade which allowed for much faster processing by discarding background regions, and hence eliminating the need to scan these areas [8].

Tansakul *et al.*, [10] used Haar-cascade classifiers for face and eye detection. They first detected the face, then, once in the face region of the image, searched for the eyes. However, they ran into issues using Haar-cascade, due to it being computationally expensive, giving slow refresh rates when compared to the speed of blink rates [10]. To solve this issue they extracted the eye region from the face image. This reduced the size of the image the tracking would need to take place on, thereby reducing computational time [10].

Method

The proposed system can be broken down into two sub-systems (Fig. 1). The front end is where the detection and counting the blinks by the driver occurs. The back end involves the comparison of blink rate and alarm trigger. Initial ideas looked at creating the back end in the analog domain to reduce power consumption whilst maintaining speed of response. The issue with this is the system requires storing of information, so the system learns the drivers steady blink rate over multiple drives, thereby improving accuracy. Therefore, it was decided that keeping the entire system in the digital domain, would simplify the system and prevent data loss when using ADCs and DACs.

The system overview is explained below:

1. A camera was used to detect the driver's eyes.
2. Frame comparisons were used to detect when the eye shuts and when the eye opens. Hence, we are checking when there is a blink.
3. Every time a blink is recorded, a high square signal is sent to the blink counter.
4. After a certain period, the counter enters its value in the shift register giving us a blink rate.
5. An average blink rate is then calculated and updated with each new blink rate calculated over a determined period.
6. Each newly calculated blink rate is compared to the most up to date average blink rate.
7. The warning system will look at the blink rate, and if it is:
   a) At the standard blink rate threshold, or higher, the alarm does not trigger.
   b) Below the standard blink rate threshold, the alarm triggers, which warns the driver they are starting to become fatigued.

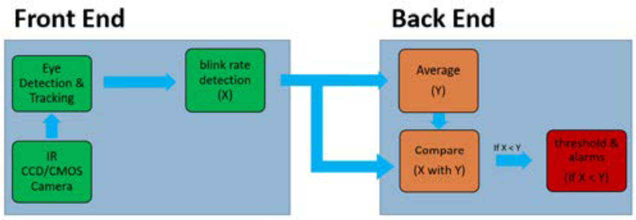

Fig: 1 overview of proposed driving awareness monitoring system

The eye and blink detection: For simulation purposes, a standard USB camera was used to detect the eyes, and then the movement of the eyelid. OpenCV [11] provides a detector based on the Haar-cascade method proposed initially by Viola and Jones [8]. The algorithm detects when the eye shuts and opens again. Every time they are detected, a high signal is sent to the blink counter which records a blink has occurred.

Blink counter: Fig. 2 shows the proposed blink-counter technique based digital counting and averaging. An asynchronous 6-bit based edge counter was designed using D-flip-flops (fig. 3a). The blink counter output, the counter state, keeps on counting to its maximum width.
When deciding which type of flip-flops to use, we looked at the limitations each type had. In general, there are four types of flip-flops; SR, JK, D and T flip-flops. JK flip flops suffer from the race around problem [12]. SR can't accept two logic high states on both inputs at a time. T flip-flops require extra hardware. While D-flip-flops are simple, glitch free flip-flops and consume less power. Hence, we ended up choosing to use D-flip-flops for our implementation [12]. D-flip-flops essentially delay the output with the same information as the input so, from a logical point of view, the data passing through is unaltered.

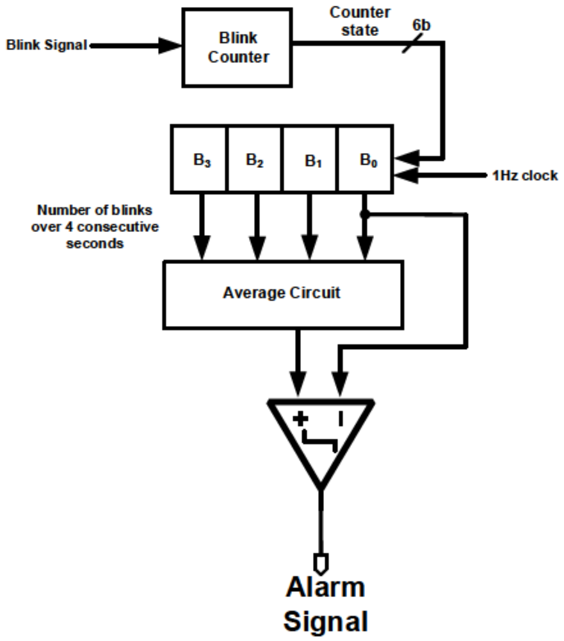

Fig. 2 working principle of Blink Rate Detection Block diagram

The blink signal is a square wave signal that goes high whenever a blink is detected. The blink counter increments by one whenever a rising edge is detected in the blink signal. Initial testing calculated the average blink rate over a period of 4 seconds. This value will likely change due to blink rate changing depending on an individual's concentration levels [5]. The counter state was fed to a 4-bit shift register (fig. 3b) with B_0, B_1, B_2 and B_3 as the information over each second. This meant B_0 contained the number of blinks in the present second, B_1 contained the number of blinks in the second before, etc.

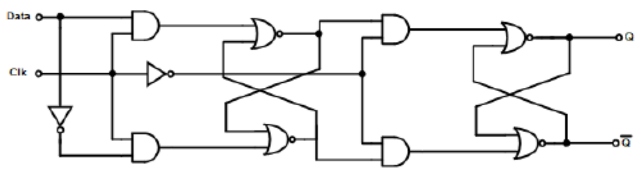

Fig. 3a schematic of each D-flip-flop used in the design.

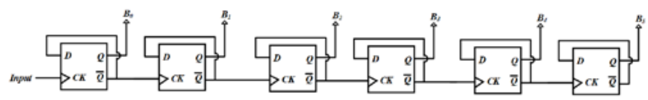

Fig. 3b schematic of the blink counter, made of D-flip-flops.

The averaging circuit output was compared with the blink rate stored in B_0 using a comparator. The output of the comparator would determine whether, or not, a signal would be sent to drive the alarm. The output of the comparator was produced by subtracting the average blink rate from the current blink rate. If the output is a negative value, then the current blink rate is higher than the average which indicates an increase in awareness and concentration levels [5]. However, if the value is positive, meaning the current blink rate is lower than the average blink rate, then a signal is sent to the alarm for activation. Note this only occurs when the difference is enough that a drop-in concentration and awareness is determined via a significant drop in blink rate. Fig. 4 shows a flow chart of the algorithm from the blink detection to triggering the alarm. Note the feedback loop when a sufficient enough difference is not detected. This is important to maintain a up to date current average blink rate which will constantly be used to check for a difference in blink rate throughout the entire journey.

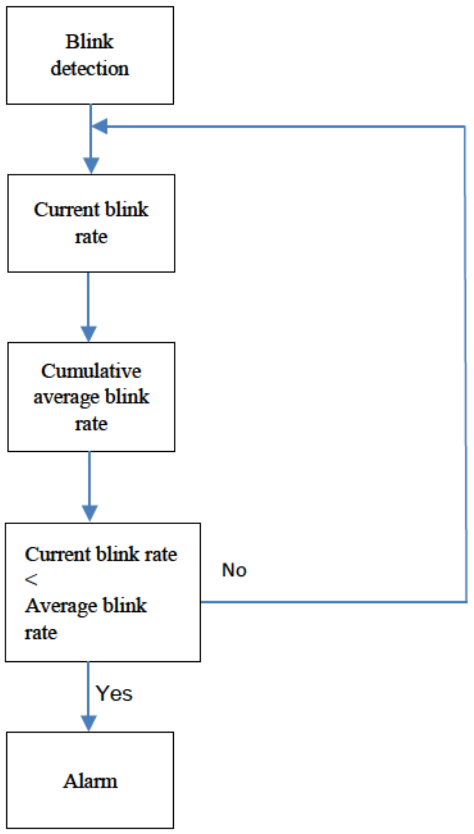

Fig 4 flow chart of proposed Algorithm

Results

Eye Detection: Preliminary results showed that Haar-Cascade Detection [7] could detect the eyes from a frontal image, and to detect the eyes open (fig. 5a) and the eyes shut (fig. 5b).

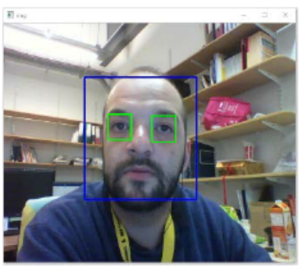

Fig. 5a eye detection of open eyes

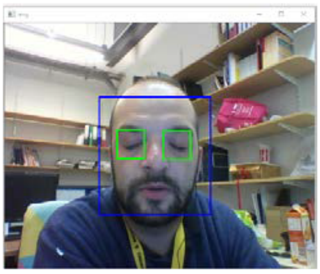

Fig. 5b eye detection of closed eyes

Initial testing found real time detection of the eyes and counting blink rates had very minimal delay in detecting the eyes.

In terms of accuracy the eye detection system could handle successful detection when the head moved 30 degrees to the left and right from starting position of looking straight at the camera. Initial testing has begun with the use of multiple cameras to look at increasing the range at which successful detection can occur.

All testing was done using normal conditions. Future testing will involve non-ideal conditions such as low light levels and the user wearing sunglasses, that obstruct the location of the eyes when suing a standard camera. This would look at illumination and occlusion limitations. This testing would involve the use of infrared cameras to observe whether these non-ideal conditions would affect the performance of the current system.

Blink Rate Detection: The system was simulated using a gate level schematic, implemented in 65nm CMOS technology (fig. 6), using Cadence software. The system consumed 130uA's from a 1.2V power supply.

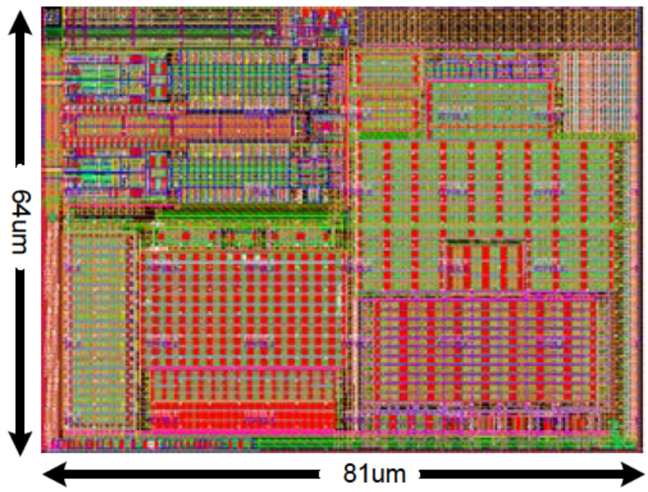

Fig. 6 simulation Layout of the 65nm CMOS technology used.

The counter described in the working principle of the counter, had a random blink signal applied to it. The output of the edge counter (fig. 7) is shown to be ramping up by one for every blink detected from the blink signal graph (Fig. 7).

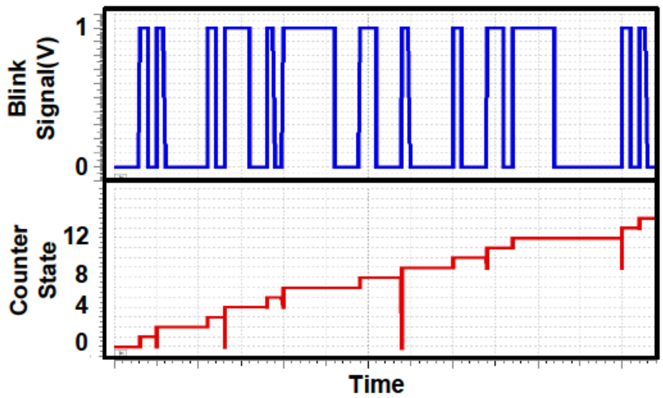

Fig. 7 working principle of the blink counter.

Fig. 8 shows preliminary results of the working principle of blink detection (fig. 2).

A random blink signal was initially applied (red trace). The blink counter keeps on updating the states and resets once it is overloaded and the alarm signal is low (Green Trace). At the end of the plot, there is no blinking in the input (meaning it is not toggling) and the resultant alarm signal is high, hence no blink has been detected and indicated.

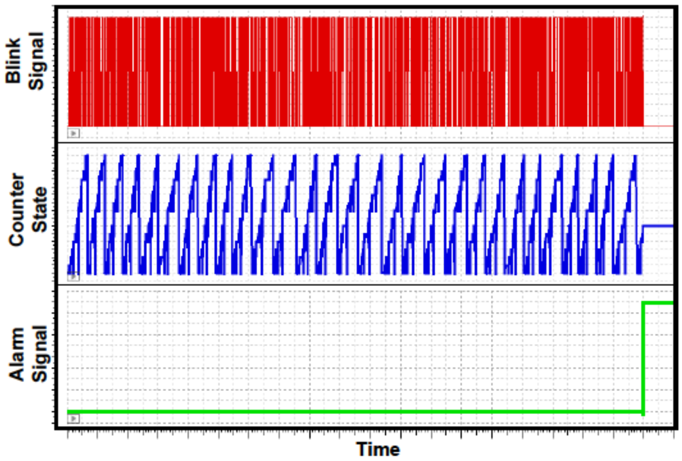

Fig. 8: Results of system simulation

During simulation runs, a few glitches occurred. There were rise and fall time mismatches among the different flip-flops, which is of minimal concern for the circuit operation, in its current state. The shift register was a serial-in, Parallel-out type and running with the 1Hz clock.

Conclusion

Preliminary results showed that a Haar-cascade method for detection of the eyes and blink rate can be used to successfully provide a digital blink signal for the blink counter. The simulation of the blink detection block shows a working principle of a current blink rate and storing multiple measurements of blink rates over some time. From this we can determine an average blink rate and compare it to the most recent blink rate, and hence send a warning signal to the alarm if necessary.

References

[1] "Driver Fatigue and Road Accidents," June 2011. [Online]. Available: http://www.rospa.com/road-safety/advice/drivers/fatigue/road-accidents/. [Accessed 3 April 2017].

[2] R. Hunter, "Staying Awake, Staying Alive: The problem of fatigue in the transport sector," Pacts, London, 2013.

[3] Optalert Adopts InterSystems Software to Prevent Fatigue-Related Industrial Accidents, Sydney: InterSystms Cache and DeepSee Case Study, 2010.

[4] R. Murray Johns and R. Christopher Hocking, "Alertness sensing device". United States of America Patent 9,007,220, 14 April 2015.

[5] Z. Ali Haq and Z. Hasan, "Eye-Blink rate detection for fatigue determination," Information Processing (IICIP), 2016 1st India International Conference on, 2016.

[6] M. Lalonde, D. Byrns, L. Gagnon, N. Teasdale and D. Laurendeau, "Real-time eye blink detection with GPU-based SIFT tracking," In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on., vol. 1, p. 1010–1017, 2001.

[7] "Face Detection using Haar Cascades," OpenCV, [Online]. Available: http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html. [Accessed 13 July 2017].

[8] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, 2001.

[9] R. E. Schapire, "The Boosting Approach to Machine Learning An Overview," Nonlinear Estimation and Classification, 2001.

[10] W. Tansakul and P. Tangamchit, "fatigue Driver Detection System using a combination of Blinking rate and Driving inactivity," Journal of Automation and Crontrol Engineering, vol. IV, pp. 33-39, 2016.

[11] "OpenCV," OpenCV, [Online]. Available: http://opencv.org/. [Accessed 12 July 2017].

[12] B. S. NAIR, Digital Electronics and Logic Design, Delhi: PHI Learning Pvt. Ltd., 2002.