ADVANCING ADVANCED PROCESS CONTROL IN BACKEND FACTORIES

Ben Williams, BS ChE, MBA
Applied Materials, Inc.
Santa Clara, CA, USA
Ben_Williams@amat.com

Todd Nowalk, BS EE Technology, MS Project Management
Applied Materials, Inc.
Santa Clara, CA, USA
Todd_Nowalk@amat.com

Chris Reeves, BS Physics, MS Education
Applied Materials, Inc.
Santa Clara, CA, USA
Christopher_Reeves@amat.com

ABSTRACT

Frontend semiconductor manufacturing has a long history of using advanced process control (APC) techniques to control chip-manufacturing processes, detect abnormalities inline, and trigger OCAP response for containment and resolution. For example, run-to-run (R2R) process control methodologies have helped fabs improve process capability (Cpk) by an average of 30% per process and increase mature yield by 2% to 4%. Similarly, fault detection (FD) has helped fabs reduce wafer scrap by up to 95% and excursions by up to 80%. While backend semiconductor manufacturing has lagged its frontend counterpart in adopting APC technologies and its operations are different from frontend, the leading-edge problems faced in wafer level packaging processes are similar and can be solved using similar approaches, and the same benefits can be realized.

For example, even though their equipment may already have hundreds even thousands of sensors, leading 300mm fabs are adding after-market sensors to their equipment to be able to detect problems that are unseen even with all the many sensors that are shipped with the equipment from the supplier. Whether it is a wafer level packaging facility or legacy technology facility, backend factories share similar challenges where some problems they are facing are not being measured or monitored. The challenge is identifying the problem to be solved, finding a sensor to detect the problem, then installing the sensor and integrating the sensor data collection with context information in a meaningful way. This paper outlines these challenges and describes methodologies to overcome them.

Key words: Run-to-Run, R2R, Fault Detection, FD, Advanced Process Control, APC, Cpk, Backend, Wafer Level Packaging

INTRODUCTION

Semiconductor demand is increasing at a rapid pace with chips being consumed in many new applications, and chip quality is also increasing as a result. For example, we have smart homes, smart buildings, public safety systems, smart vehicles, smart watches, IoT, and social media; and many of these applications include numerous uses that have surfaced over the last decade. It is anticipated that a city of 1 million people will generate 200 million GB of data per day by 2020 [1]. As our dependence on semiconductors has increased, the quality demands on semiconductor manufacturers has also increased significantly over the last decade—especially when looking at applications associated with automotive where the average number of microprocessors has risen from about 15 microprocessors in 1999 to 50, to 100, and even up to 150 microprocessors in some of today's cars (note this is only microprocessors and does not include other chips) [2][3]. With this increased dependence on automotive applications, it is evident that chip failure can potentially be fatal. If total car production in 2017 was nearly 100 million cars worldwide [4] and the average number of microprocessors is close to 50, then a defect rate of ppm would mean thousands of potentially catastrophic failures. *Zero defects* is a common refrain among today's semiconductor manufacturers, and it is no wonder.

Because of the higher quality demands being placed on semiconductor manufacturers, a subsequent push exists for improved quality business processes and supporting software systems. In fact, this push extends across the entire value chain from supplier to fab and to assembly, test, and packaging as illustrated in Figure 1. Quality business processes working together with a continuous feedback loop that moves from customer to backend factory to fab to raw materials supplier are necessary to improve overall quality.

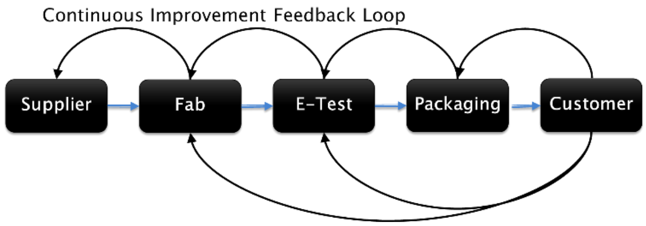


Figure 1. Continuous Improvement Feedback Loop

Moreover, providing automation systems and tools to facilitate that feedback loop is critical to enabling even higher quality.

Frontend semiconductor manufacturing has a long history of using advanced process control (APC) systems such as fault detection and classification (FDC) and run-to-run (R2R) to detect abnormalities inline and to trigger OCAP response for containment and resolution as well as to control the chip-manufacturing processes. Specifically, 300mm fabs have benefitted from high levels of automation that have facilitated data collection for related APC applications. As frontend fabs have enjoyed investment dollars and R&D with these applications, backend factories (and technologies older than 300mm) have not received the same level of investment, which has created some significant challenges to adopting these APC applications and receiving the related benefits.

As we have met with backend customers worldwide, we have noticed some general themes with respect to APC. We see the main obstacles to adopting FDC and R2R are standardization, sensor selection, merging data streams, data collection rates, data aggregation, recipe times, statistical knowledge, and process control knowledge. The lack of standardization causes problems with tool automation and automated data collection. Sensor selection, merging data streams, and data aggregation are about finding the right signal to detect problems and merging it with relevant context information so that the data becomes useful for detection. The short recipe times mean that data collection frequency must be high to capture the deviations of interest. Regarding the last two items, this is not to say that backend factories do not have this knowledge base; rather, we have noticed a higher concentration of engineers with these skills who specialize in FDC, R2R, and SPC in fabs. In this paper, we outline some methods and practices that have proven to be beneficial in frontend, backend, display, and even non-semi industries.

THE QUALITY PROCESS

Deviation Review and FMEA Highlight Detection Needs

Assuming higher quality is a manufacturing goal, we realize that detecting deviations is a central concern. One question we commonly hear is "Where do I start?" and the answer is to use failure modes and effects analysis Failure Mode Effects Analysis (FMEA), root cause analysis, and deviation review help prioritize efforts (8D, 5-why, etc.). Our objectives in the quality process are to uncover potential points of failure (FMEA), perform root cause analysis (8D, 5-why), identify detection points, isolate clear signals that prevent or limit a deviation and its impact, and ultimately to implement safeguards to prevent recurrence (8D). In the quality process FDC plays a preemptive and pivotal role, and its sole purpose is to provide detection.

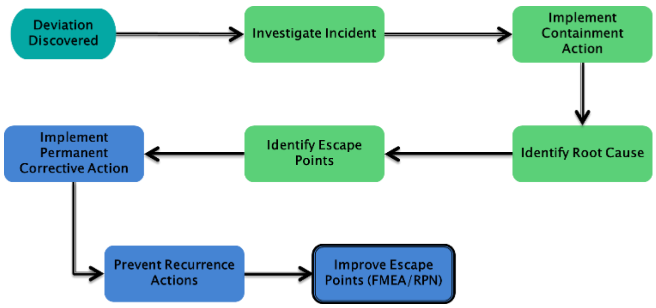
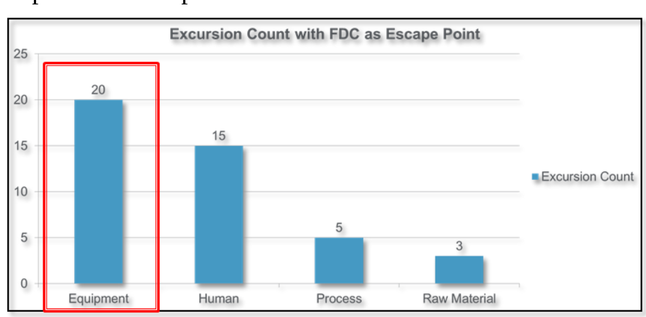


Figure 2. Simplified Deviation Review Process

One output of the FMEA is the Risk Priority Number, or RPN, which is a product of three characteristics of the failure mode: Severity, Occurrence, and Detection (S x O x D = RPN). As noted earlier, FDC and SPC provide detection, and failure modes where D is high, meaning the detectability is low, should be prioritized to find some signal to prevent or limit deviations. Figure 2 shows a simplified deviation review process in which the final step in the process is "Improve Escape Points." If we use FMEA/RPN, then the work to implement or improve FDC should be prioritized in that step.

Another important step in the deviation review process is "Identify Escape Points." In the 8D parlance this is the closest point in the manufacturing process where root cause could have been found (think detected) but was not [6]. Using pareto methodology shown in Figure 3 and Figure 4, we can categorize and sub-categorize our escape points in a way that allows us to identify the most fruitful sectors to implement or improve FDC.



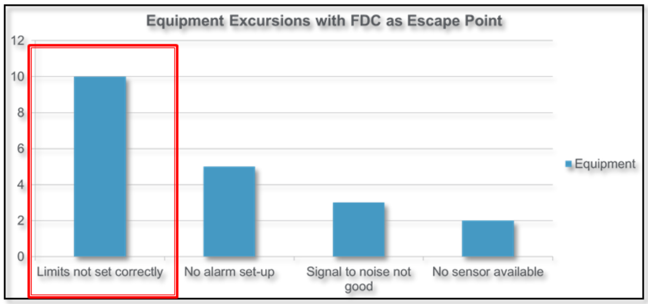


Figure 3. Example Deviation Pareto

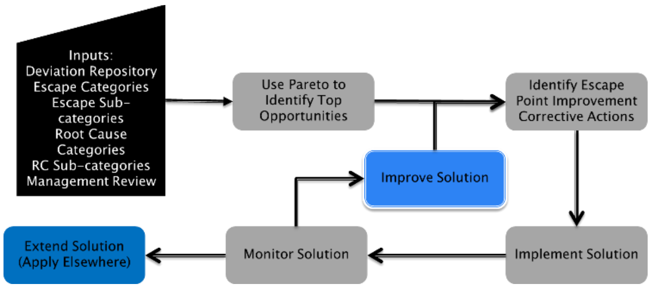


Figure 4. Example Escape Point Improvement Process

FAULT DETECTION

Tool Connection Types and Sensorization

Once the key areas for improvement have been identified, the FDC implementation and improvement activities can begin. As noted in the introduction, these activities are especially challenging for backend factories due to lack of tool communication standards, tool automation, and data collection. Additionally, many processes/recipes in backend manufacturing are very short relative to what frontend fabs see, which necessitate high data collection frequencies for some processes. This section describes some of the various methods used to collect tool and sensor information from backend equipment and import them into a fault detection system.

Table 1 outlines various data connection types that are used in semiconductor manufacturing along with the relative ease in connecting such data types to FDC. SECS/GEM and Interface A are commonly used, easy to connect, and relatively plug-and-play, so this paper does not discuss the use cases associated with these two types; instead, it focuses on the use cases for OPC, TCP/IP, flat file/binary file, and ODP. These connection types are not as plug-and-play as the connection types mentioned previously; however, they are commonly used successfully in semiconductor and other industries that are using FDC.

Table 1. Data Connection Types

Connection Type	Relative Ease	Comment
SECS/GEM	Easy	Most common connection in semi front end. Shared connection with equipment automation program.
Interface A	Easy	Second most common connection. Ability to host multiple applications.
OPC	Easy	Used in industrial and subfab applications, PLCs, etc.
TCP/IP	Easy	Needs DLL for each connected device.
Flat File/Binary File	Easy	FDC supports via ODI.
TIBCO	Easy	FDC driver available.
ODP	Moderate	Data systems protocol. Requires custom DLL.

OPC Connections

OPC is a communication protocol standard that enables cross-platform data sharing independent of the software or supplier. Typical data sources using OPC are PLC and real-time data application (OPC DA) such as mass flow controllers (MFCs) and gas boxes, temperature sensors, pumps, chillers, heater jackets, and abatement equipment. In addition to the real-time data, other data such as alarms and events (OPC AE) as well as historical or archived data (OPC HDA) are sent using OPC. Figure 5 shows a hierarchy of data being sent to factory applications through OPC. Fault detection connects to the OPC server to receive the data. These collection rates can satisfy the need for high frequency data.

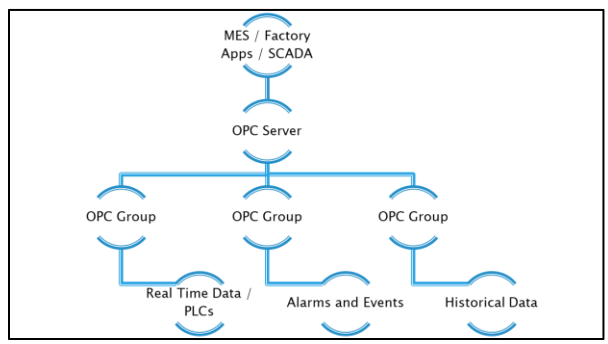


Figure 5. OPC Protocol

TCP/IP Connections

Transmission Control Protocol (TCP) divides a message or file into packets that are transmitted over the internet and then reassembled when they reach their destination. Internet Protocol (IP) is responsible for the address of each packet so it is sent to the correct destination. Sensors utilizing TCP/IP can also provide data at the high frequencies needed to detect problems on short processes. The following two examples illustrate the ease of setting up the TCP/IP connection between the sensor and FDC system.

In the first example, we had a very cost-conscious customer who wanted to measure the wafer's surface temperature at a specific process step, but the tool as configured did not support such a measurement. The solution needed a cost-effective proof of concept followed by a small-budget deployment because the customer had several hundred tools requiring this configuration.

The customer procured a specialized sensor to monitor the surface temperature and needed help to connect the sensor to FDC and merge its data into the main data stream of the tool. To complete the connection the following items were needed: Lab Jack U3HV, Raspberry Pi box, data acquisition (DAQ) adapter written in Python, and a plugin DLL written in C#. Refer to Figure 6 for a schematic of the configuration. Ultimately, the proof of concept was successful, and the customer was able to deploy the solution to the several hundred tools in its fleet.

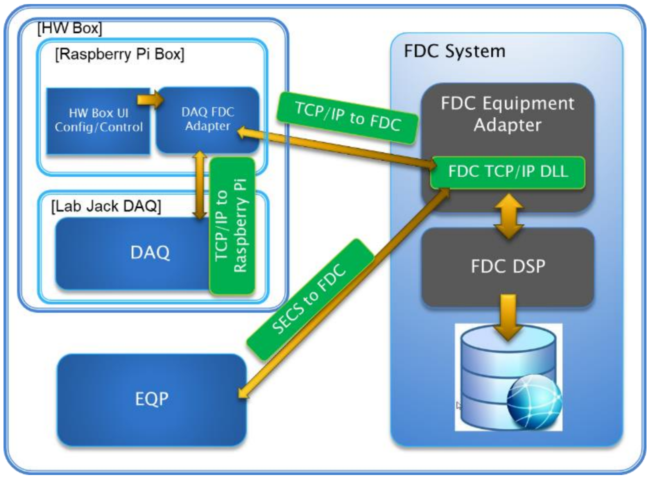


Figure 6. TCP/IP Connection Using Raspberry Pi

The customer could have used a less complicated direct connection from the DAQ to the FDC system; however, indirect Raspberry Pi was selected because it was less expensive than buying a server to support a direct connection from the DAQ to the FDC system. A Raspberry Pi box costs $35 versus a comparable mini PC at about $300. Given the size of the fleet, the customer was able to save over $100,000 by using the indirect connection through the Raspberry Pi.

The second example of TCP/IP connection was provided by IMEC in Leuven, Belgium. Similar to the first example, the customer added a sensor to the tool (Laytec EPI-TT temperature sensor) and needed to merge the sensor's data with the main FDC data stream coming from the tool itself. The main difference between this case and the first case is how the sensor was integrated. In this case, the customer used a less complicated direct connection from the sensor to the FDC system.

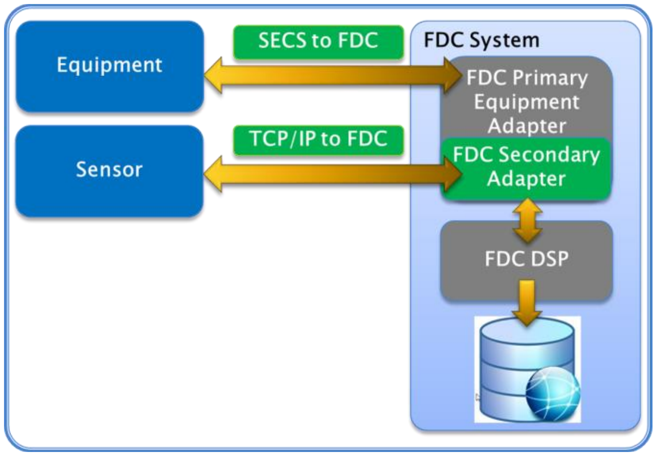


Figure 7. TCP/IP Connection Using Secondary Adapter

Flat Files & Binary Files Using ODI Connections

In some instances, the equipment is either incapable of a SECS/GEM or Interface A connection or the tool is capable, but the connection is unreliable. In such cases, it is imperative to find a way to collect the data reliably. Some tools can generate files of data and store them. Depending on the tool, the internal data collection frequencies can reach high into the kHz range. In such cases, we recommend using offline data integrator (ODI) to send the data to FDC.

The ODI retrieves the files from their stored location, parses them into a language understood by the FDC system, and finally merges the files with other context information as it is used in the FDC system for calculation and analysis. This method is common, and we have seen customers use it in backend, frontend, and non-semi with success.

As mentioned earlier, one advantage of ODI is the high data frequency. One backend customer chose to use ODI as the primary medium because the deviation they were trying to detect mandated higher frequencies than what SECS/GEM was able to provide.

At a frontend fab, we connected some ashing tools to FDC using this ODI method. In this case, ODI was the medium of choice because of the low data quality coming from the primary SECS/GEM connection. The root cause was the equipment interface (EI) had instability issues that were preventing reliable data collection.

The solution was to use the flat files produced by the tool and import them into the FDC system. The FDC system has a build-in ODI loader, so it was easy to set-up the connection. The FDC system is also able to collect the files and import them into FDC near real-time—meaning that it can import files and perform analysis as if the FDC system were polling the tool data from SECS/GEM and performing analysis at the end of the run.

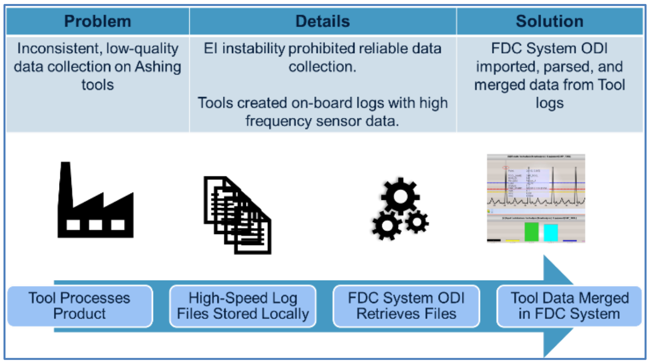


Figure 8. Offline Data Integrator

ODP Connections

Though perhaps not as common as other data connection types, Open Data Protocol (ODP) is an available option if FDC needs to query an existing data source. Both semi and non-semi customers are using this method. Microsoft initiated ODP (referred to as OData), and other flavors are available such as Google's GData. ODP "allows the creation and consumption of queryable and interoperable RESTful APIs" [7]. Typical uses include querying databases and applications such as: SAP NetWeaver Gateway, MS SharePoint, Office 365, Tableau, and Tibco Spotfire.

Sensorization

The quality section previously described a process for determining priorities for implementing FDC for the purposes of detecting anomalies (FMEA and RPN). It also described a process and method for making improvements to FDC (escape point improvement). As FDC has been adopted, data consumption has exploded—partly because manufacturers looking for more data or faster data to detect problems have added sensors to equipment as one way to achieve that goal.

As factories seek to add sensors, a phased approach in doing so is essential. The first phase is to create the data merge (i.e., many streams to one), and this includes both analog and digital I/O support. The second and third phases are implementing commonly used sensors—essentially borrowing best known methods from the industry. Lastly, the fourth phase is moving on to more advanced sensors such as acoustic, vibration, RF probes, and the like. Plug-and-play capabilities are important to reduce the time-to-solution to include development and deployment costs.

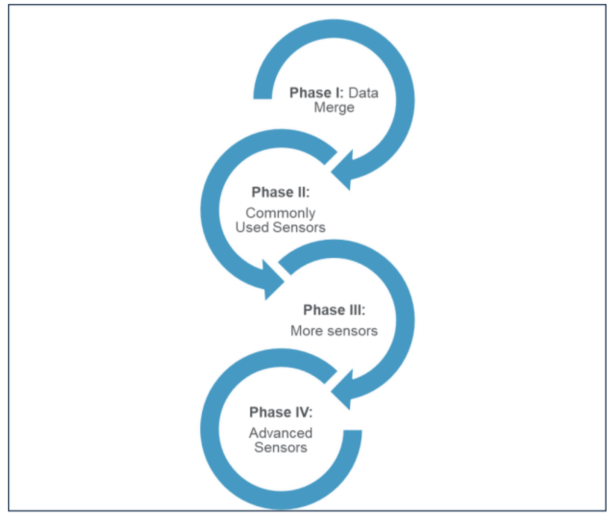


Figure 9. Phased Approach to Sensorization

Statistical and Process Control Expertise

As mentioned previously, FDC, R2R, and SPC are pervasive in both frontend and display fabs and are gaining a foothold in backend factories. Even though these applications are pervasive, not all factories are equally effective in how they implement them.

One common thread among factories that have successful APC programs is they have dedicated teams who specialize in process control. Centralizing this knowledge and skill serves two purposes. First, APC applications require a unique skillset serving a unique function. Second, centralization facilitates consistency, standardization, and promotes sharing of best known methods. Just as product engineering requires a special skillset or process engineering also requires a unique skillset, controls engineering is also specialized. Quality demands consistency and control.

Though the following list is not exhaustive, nor is it exclusive, it serves as a general guideline for building the expertise required to manage a successful APC program. A successful controls engineer acts as a business analyst. He or she must understand the processes and equipment that must be controlled. An understanding of IT is also critical so he or she can understand the control problem that customers face, design a solution that meets the customer's need, and work with IT (if needed) to implement the solution.

In general, good controls engineers have scripting or programming ability, though the software they use may or may not require programming. He or she also has good analytical skills, is able to use different algorithms to solve different problems, and understands statistics and SPC. With R2R, having a background in controls or robotics is especially useful in understanding control theory as it relates to semiconductor processing

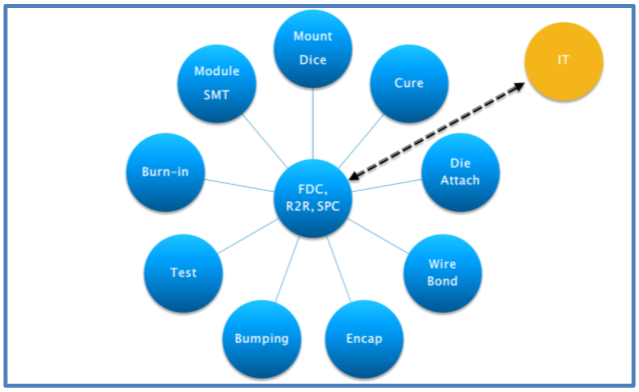


Figure 10. Example Organization with Centralized APC

CONCLUSION

Semiconductor manufacturers are experiencing increasing quality demands. With IoT, mobile, and automotive applications all using an increasing number of microchips and producing and consuming more and more data, the quality requirements of microchips is increasing, and companies are looking for more advanced ways to improve quality.

More and more backend fabs are adopting APC applications like FDC and R2R. Along with adopting such technologies, these fabs are experiencing challenges. Some problems are similar to those faced by sectors already using APC, and other problems are unique to backend operations. Learning what works and why in other sectors (as well as what has failed and why) will prove beneficial to companies looking to implement FDC and R2R. Some challenges can be addressed by existing solutions and methods while others will require continued investment for resolution.

Having a continuous improvement feedback loop that spans the entire value chain is beneficial to everyone involved and is critical to improving quality. Additionally, using a pragmatic process to identify opportunities and prioritize work for FDC will help guide companies in adopting FDC. FMEA, RPN, 8D, and 5-why among other methods should be used to help direct the work of deploying FDC. As companies incorporate FDC into their factories, they should use the various methods to collect data as they are applicable. Adding sensors to tools is a great way to detect critical problems. Leveraging existing tool data stored in flat files is another way to collect quality data that facilitates detection. Supplementing data with outside sources can enhance the dataset and improve the analytical results.

Finally, investing in APC applications does not end with purchasing or developing an APC system. Companies should seek to develop or hire people who possess the requisite skills to successfully drive the program forward. Building teams and centralizing the skill is common among companies with effective programs.

ACKNOWLEDGEMENTS

The authors would like to thank Ray Murray, Sung Kun Park, Rick Stafford, Ryan Sealy, Albert Qiu, Edris Safari, Damon Tai, and Alex Fan from Applied Materials for their help and contributions. Additionally, we would like to recognize Vincent Nollet and Kristof Coose from IMEC for allowing us to highlight their successful TCP/IP use case.

REFERENCES

[1] Gary Dickerson, “Innovation Leadership in Expanding Markets,” Applied Materials, Inc., Message to analysts, September 2017. [Online]. Available: http://edge.media-server.com/m/s/aw5vyj3m/lan/en. [Accessed September 14, 2018].

[2] Jim Turley, “Embedded Processors by the Numbers,” *EE Times*, May 1, 1999. [Online]. Available: https://www.eetimes.com/author.asp?section_id=36&doc_id=1287712. [Accessed September 13, 2018].

[3] Paul A. Eisenstein, “Alexa, Start My Car: How Next Year's Models Are in the Fast Lane for High Tech,” *NBC News*, October 9, 2017. [Online]. Available: https://www.nbcnews.com/tech/tech-news/alexa-start-my-car-how-next-year-s-models-are-n809036. [Accessed September 14, 2018].

[4] Statista, The Statistics Portal, “Worldwide Automobile Production from 2000 to 2017 (in million vehicles),” *Statista, The Statistics Portal*, 2018. [Online]. Available: https://www.statista.com/statistics/262747/worldwide-automobile-production-since-2000/. [Accessed: September 21, 2018].

[5] Doug Parker and Christopher Thomas, McKinsey & Company, Inc.
“Winning Share in Automotive Semiconductors,” *McKinsey on Semiconductors*, Autumn 2013. [Online]. Available: https://www.mckinsey.com/~/media/mckinsey/dotcom/client_service/semiconductors/issue%203%20autumn%202013/pdfs/5_automotivesemiconductors.ashx. [Accessed: September 18, 2018].

[6] “Open Data Protocol,” *Wikipedia The Free Encyclopedia*, [online document], Available: Wikipedia, https://en.wikipedia.org/wiki/Open_Data_Protocol [Accessed: September 14, 2018].

[7] Quality-One International, “Eight Disciplines of Problem Solving (8D),” *Quality-One International*, 2018. [Online]. Available: https://quality-one.com/8d/. [Accessed: September 12, 2018].