Seminar Paper: Battery-Free Game Boy

Florian Koch
Friedrich-Alexander-Universität Erlangen-Nürnberg

ABSTRACT

Relying on batteries as power sources can often be undesirable due to their contribution to the increased size and weight of electronic devices, as well as the inconvenience associated with charging. Moreover, the batteries themselves and the energy required for recharging are often environmentally unfriendly. However, attempting to operate electronic devices without a consistent and stable power source poses significant challenges.

This paper explores the design of a deeply interactive device, specifically a Game Boy, to achieve energy self-sufficiency without relying on a battery. The investigation involves examining various methods for power generation, enhancing the device's energy efficiency, and introducing a novel system called MPatch to address memory loss caused by power outages. Subsequently, the paper presents test results and evaluates the gaming console based on these findings.

1 INTRODUCTION

In an era marked by the urgent global challenge of climate change, the intersection of technology and environmental responsibility becomes a focal point of paramount importance. The proliferation of electronic devices in our everyday lives not only results in increased energy consumption, contributing to a higher CO2 footprint, but also underscores the pressing need for greener energy sources. Furthermore, the devices themselves could operate much more energy efficient[5]. The reliance on batteries, a common power source for these devices, introduces additional detrimental effects on our planet. Beginning with the extraction of necessary raw materials, the mining process often leads to habitat destruction and deforestation. Moreover, the manufacturing process is extremely energy-intensive, and batteries contain hazardous materials that pose a significant risk to the environment upon disposal[7]. Unfortunately, recycling is infrequently practiced due to its inherent difficulty. Instead, batteries are typically either deposited in large landfills or incinerated. Addressing these issues is imperative for humanity to preserve a sustainable and undamaged planet.

One factor that could be optimized involves minimizing reliance on batteries and instead designing devices to be energy self-sufficient, utilizing the various forms of energy that are omnipresent, such as light, heat, or motion. Particularly for compact devices like phones, watches, or handheld consoles, which have relatively modest energy requirements, exploring self-sufficient power solutions holds significant promise. Eliminating the need for batteries also presents additional advantages, as it reduces both the size and weight of the device, a practical consideration for handheld devices. The absence of battery dependence eliminates the necessity for regular charging, a considerable advantage for devices intended for on-the-go use.

A pioneering step toward interactive, battery-free intermittent systems with demanding memory preservation needs was taken by Jasper de Winkel et al. with their creation of the battery-free Game Boy named ENGAGE. This innovative device aims to deliver an authentic gaming experience even during power outages, ensuring that players can continue their game without having to restart. The gaming industry continues to expand, and handheld gaming remains significant, as evidenced by the remarkable success of the Nintendo Switch, one of the most successful consoles in history[9]. Consequently, the ecological impact of such gaming devices cannot be overlooked.

The following section introduces the ENGAGE Game Boy, along with the challenges encountered during the design process and their respective solutions. The utilization of solar panels and mechanical switches[1] for power generation and other alternative approaches employed in similar devices are investigated. Additionally, the incorporation of an incorruptible and efficient checkpointing system named MPatch is being presented and compared to competing solutions. The final sections evaluate the device and discuss the results derived from the work of Jasper de Winkel et al..

2 BATTERYLESSNESS AND ITS CHALLENGES

2.1 The Nintendo Game Boy Emulator

Jasper de Winkel et al. have designed a Nintendo Game Boy emulator employing modern hardware components. The primary objective was to deliver an authentic gaming experience, minimizing compromises wherever possible. This entailed ensuring the emulator's compatibility with all games featured on the original platform. In the event of a power failure, the emulator was engineered to resume the game from the exact point of interruption upon restarting, eliminating the need to start the game anew—a crucial feature for facilitating uninterrupted gameplay. Maintaining the original aesthetic was also a priority, requiring the emulator to replicate the size, buttons, and the overall front and back design of the original Game Boy with the addition of solar panels.

2.2 Challenges

The absence of a battery and the circumstances surrounding the device present significant challenges:

- **Fluctuating power supply:** Without a battery, the system cannot store excess energy, leading to shutdowns during phases of low available energy.
- **Game independence:** Games vary significantly in their interaction with users and hardware/software of the Game Boy. To support all games without optimizing for a single one poses a challenge.
- **High system requirements:** Games, in general, demand substantial computation, placing significant strain on the system.
- **Expensive checkpointing:** Due to frequent power outages and the need to save the game state, large amounts of

volatile memory must be transferred to non-volatile memory multiple times, intensifying stress on the computing system and increasing energy requirements.

These difficulties should be overcome while ensuring smooth gameplay.

2.3 Game Boy Workflow

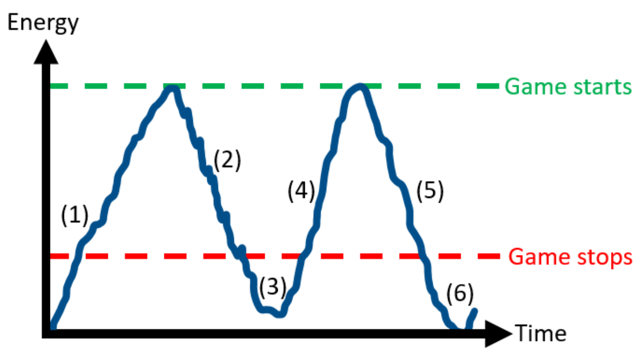

Figure 1: Different phases of the Game Boy depending on the available energy

Figure 1 illustrates the Game Boy's operation based on its current energy levels. The Game Boy is powered by solar panels, resulting in lower energy production during periods with less sunlight (1) and significantly higher production during phases with ample sunlight (4). The Game Boy undergoes a "charging" phase until its energy capacities are full, at which point the **game starts**. While playing, additional energy is generated through mechanical switches activated by button presses (2). Subsequently, during videos with no input demand, no energy is produced in this manner (5). The **game stops** as the device approaches power failure. Only during low-power phases, checkpoints are generated. Therefore, energy continues to be consumed even after the game has stopped (3). In instances where energy depletes before a checkpoint is completed (6), there is a potential risk of encountering "corrupted" checkpoints.

The subsequent sections delve into the underlying concepts of the Game Boy and explore the strategies employed to achieve a properly functioning device.

3 ENERGY PRODUCTION AND CONSUMPTION

3.1 Energy Self-Supply

In the realm of generating our own green energy on the go, it is crucial to have a reliable energy source or diversify across multiple sources that complement each other. This ensures the ability to play under various circumstances. We have three forms of energy—light, heat, and motion—that can be transformed into electricity in different ways.

For light, simple **solar panels** can be utilized to generate a portion of the energy, similar to their use in calculators today.

In terms of heat, **thermoelectric generators** (TEGs) have been successfully employed for energy self-sufficient portable devices, such as in powering watches [10]. TEGs leverage the Seebeck Effect, where two different metals or semiconductors connected in a closed circuit create a voltage potential when there is a temperature gradient across them. Electrons in the hotter region gain energy and move towards the cooler region. Nguyen Van Toan et al. have applied this principle to use the temperature difference between the human body and the surrounding ambient temperature to power a watch, a concept that could be easily adapted for the Game Boy.

Using motion as a proven method of energy supply suitable for the gaming console, **piezoelectric generators**[6] emerge as a viable option. These generators employ piezoelectric materials that deform under mechanical stress, leading to the separation of positive and negative charges. The resulting electric potential across the material is then collected as an output voltage. In the context of the Game Boy, vibrations created through button pressing or shaking the device could be harnessed to generate power.

Mechanical switches and **solar panels** are the sole energy supply of Engage, making the device highly dependent on access to light and games that involve frequent button pressing. Specifically, the front of the Game Boy is covered with generic solar panels, and off-the-shelf mechanical switches[1] are used for the A and B buttons as well as the D-Pad. These switches operate by moving a magnetic block from top to bottom upon pressing, with the abrupt reversion of polarity transforming mechanical energy into electrical energy. Upon release, the block returns to its original position. To maximize power generation, a button can only be held for up to 300ms before needing to be pressed again to signal an input to the game, slightly altering the gameplay for certain games.

Regardless of the energy source, minimizing power consumption is a top priority for increasing the device's uptime.

3.2 Reducing the Needed Energy

To minimize energy consumption, modern hardware components have been carefully chosen for the Game Boy. An Ambiq Apollo3 Blue ARM Cortex-M4 MCU, an ultra-low-power microcontroller featuring Flash and SRAM, is being deployed. Additionally, an external Fujitsu MB85RS4MT 512 KB FRAM is incorporated for fast, low-power, non-volatile memory essential for storing checkpoints. The Japan Display LPM013M126A LCD is selected for its easy availability and, once again, its low power consumption.

Compromises have been made in the area of sound, as it is not a feature of the Game Boy emulator. This limitation is partly attributed to the challenges associated with creating an enjoyable sound design for an intermittent device.

As previously mentioned, constant checkpointing poses a significant challenge in terms of power consumption. Memory must be moved frequently, and some level of overhead is inevitable. Optimizing this process represents an intriguing research topic that could potentially yield promising results in enhancing energy efficiency.

4 MEMORY MANAGEMENT WITH CHECKPOINTS

4.1 Categories of Checkpointing Systems

All checkpointing systems can be divided into two categories:

- **Corruptible checkpointing:** The current state of the MCU is transferred to a predetermined location on non-volatile memory, overwriting the old checkpoint. This process may lead to a corrupted checkpoint if the current one is not completed, while parts of it have already overwritten the previous one. Ensuring that sufficient energy is present for a new checkpoint to be completed with 100% certainty is unrealistic, as predicting the energy needed for a checkpoint in a complex system like a battery-free Game Boy is practically impossible.
- **Incorruptible checkpointing:** These checkpointing systems operate by not directly overwriting a previous checkpoint, ensuring that there is always a 100% functional checkpoint available.

For playing a game with an intermittent power source, using an incorruptible checkpointing system is crucial; otherwise, the Game Boy would frequently generate broken checkpoints, and the game would have to start over. Now that we know an incorruptible checkpointing system is necessary, the question remains as to which one is best suited for our use case.

4.2 The Problem with Power-Efficient Incorruptibility

An approach to achieve incorruptible checkpoints involves using two buffers—one for the current checkpoint and one for the previous one. The alternation between which buffer is overwritten after every successful checkpoint ensures that if a checkpoint fails, the older one can be used to restore the game state. However, this solution may not be optimal in terms of energy efficiency, as an older checkpoint might partially contain the same information as the current one, resulting in overhead as already available information is saved again.

Saad Ahmed et al. explored a differential checkpointing technique for intermittent systems[2] that tracks modifications in memory and only saves the altered memory, significantly reducing overhead and power consumption. This approach is particularly promising when only a few parts of the memory are frequently changing. Jasper de Winkel et al. tested this for gaming by tracking memory changes in four different Game Boy games. The result showed that significant portions of the memory were changing rarely, making differential checkpointing well-suited for the Game Boy.

The system developed by Saad Ahmed et al., called DICE[3], is a corruptible one, meaning it needs transformation. However, simply applying differential checkpointing to a double buffer doesn't work because with differential checkpointing, a checkpoint always requires information about the preceding checkpoint, creating a dependency between the two buffers. This issue could then only be resolved by introducing data duplicates again.

4.3 MPatch

To address this challenge, Jasper de Winkel et al. developed the first incorruptible, differential checkpointing system called MPatch. The name is derived from the term "patches," which represent consecutive segments of volatile memory, specifically the modified memory between checkpoints, saved onto non-volatile memory. These patches are linked through a linked list, where each patch refers to the next younger patch. Each patch is assigned the number of the checkpoint it corresponds to. After successfully creating a checkpoint, an atomic variable n is incremented, indicating that all patches with a number less than n can be utilized for game state restoration in the event of a power failure.

In detail, when a power failure occurs and the system needs to restore its state upon restarting, the linked list is traversed from the newest patch to the oldest patch. The contents of patches with a number equal to n pertain to an incomplete checkpoint and are disregarded, and the checkpoint itself is promptly deleted.

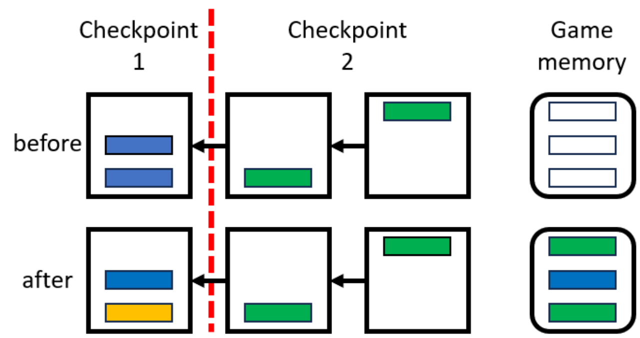

Figure 2: Game memory directly before and after restoration

Figure 2 illustrates how the saved memory from the remaining patches is utilized to reconstruct the checkpoint. For each patch, only the segments of the saved memory that have not been already applied in a newer patch are used, ensuring that older information does not overwrite newer data. An interval tree is employed for this purpose, recording the memory range that has already been altered. Consequently, for checkpoint 2, the entire memory can be copied to the Game memory. For checkpoint 1, however, parts of the memory are outdated (indicated in yellow) as checkpoint 2 contained newer data at the same location in the memory. Consequently, this patch is only partly applied. The restoration process concludes after traversing the entire list, and the game can resume.

This approach has the dual advantage of being both non-corruptible and differential. While there are no duplicates of data, some patches may contain unnecessary memory that will never be applied—an inevitable outcome. Importantly, there is no memory overhead at the moment a patch is created, ensuring that the Game Boy is not burdened with unnecessary computational load and energy drain during gameplay, thereby enhancing the gaming experience.

4.4 The Memory of ENGAGE

The emulated game memory utilizes the recently introduced MPatch system. The remaining emulation management logic is checkpointed using a straightforward double buffering method. Unlike the original Game Boy, the screen has its own buffer that is also checkpointed to facilitate the restoration of its specific state.

The following section presents the performance of the device during actual testing by first examining the uptime of the device and then comparing MPatch to an alternative.

5 EVALUATION

5.1 Power Outages Throughout Testing

As demonstrated in the preceding chapters, the device's performance is highly contingent on the game being played and the availability of light. Consequently, four different games were tested: Tetris, Space Invader, Super Mario Land, and Bomberman.

In terms of average button presses per second, Bomberman exhibited the lowest value at 1.3 presses per second, while Space Invaders had the highest at 2.7 button presses per second. This is only an average and can vary significantly depending on the player or the current game situation, for instance, cutscenes temporarily reduce button presses to almost zero. One button press per second generates 0.66mJ. Rounding the test results for different games yields 0.66mW in the worst-case scenario and 1.97mW in the best-case scenario. Jasper de Winkel et al. assessed performance under two different lighting conditions, with strengths of 40/20 kilolux. The first scenario yielded an average of 10.14mW, while the latter produced 8.33mW.

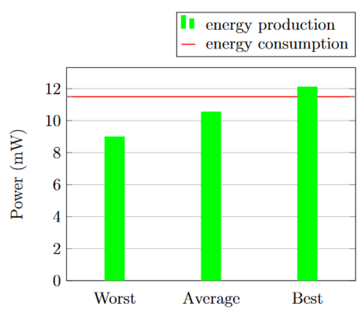

Figure 3: worst, average and best case energy production vs consumption

Figure 3 illustrates the power production under these conditions in comparison to power consumption. ENGAGE, on average, **consumes 11.5mW**, representing a substantial improvement compared to the original Game Boy, which consumes 232.08mW during game execution. However, in the **worst-case** scenario (1 button press per second + 20lkx -> 9mW) and non-ideal situations, ENGAGE consumes more energy than it generates, leading to an inevitable power outage. Even in the **best-case** scenario (3 button presses per second + 40klx -> 12.11mW), there are phases of lower button press rates and reduced energy generation through the solar panels, resulting in occasional energy depletion. The **average-case** scenario, calculated based on the total power of the best- and worst-case scenarios, also yields less power (10.55mW) than ENGAGE consumes. Tetris, with an average of two buttons pressed per second, has a gaming uptime of approximately 10 seconds with 40klx or 3.5 seconds with 20klx, underscoring the relevance of a good light source.

5.2 MPacth vs Non-Differential Double Buffering

Other tests conducted compared MPATCH in terms of the time it took to create and to restore a checkpoint against a naive checkpointing approach similar to MEMENTOS[8], which is based on non-differential double buffering. In these tests, MPATCH demonstrated superiority in both aspects, significantly reducing the time required for checkpoint generation and slightly decreasing the time needed for checkpoint restoration. As a result, MPATCH appears to successfully fulfill its purpose of reducing overhead and, consequently, lowering the power consumption associated with checkpointing.

6 DISCUSSION

Improvements in terms of power consumption and checkpointing efficiency have undoubtedly been achieved. However, in any testing scenario, the gaming experience remains unsatisfying. Furthermore, testing under actual absence of light was not conducted, but the results would have been predictably poor, given that ENGAGE relies almost entirely on solar panels. With the additional use of other energy sources, such as deploying a TEG, more favorable results could certainly have been attained. Even then, it's essential to acknowledge that the built device is an 8-bit Game Boy, with modern gaming consoles having significantly higher energy demands. It is important to note, though, that Jasper de Winkel et al. never claimed to have developed a device for practical application in the real world. Instead, their aim was to demonstrate the **possibility** of enabling battery-free intermittent gaming, a goal they certainly achieved. Particularly, the developed MPatch system is a genuinely interesting and promising concept that has been successfully validated through testing.

However, the necessity for gaming consoles to be battery-free is questionable. Convenience is a weak argument, especially considering how rapidly modern devices can charge and the widespread accessibility of electrical sockets. A console that consistently shuts down cannot be considered convenient either, which is the biggest challenge when designing a battery-free console. An important area of research focuses on making batteries more environmentally friendly[4, 11]. This may be a more relevant emphasis for video game consoles, considering the ecological impact, as there are no other compelling reasons to entirely abandon batteries in this context.

The situation is different for ultra-low-power devices or devices where relying on a battery would introduce significant inconvenience, such as medical body implants, for which abstaining from batteries is likely to be the future[12], as changing the battery in such devices involves a substantial effort.

7 CONCLUSION

All in all, it can be concluded that the developed Game Boy does indeed function, although the current version is only suitable for demonstration purposes in a laboratory environment, but future advancements in energy efficiency and energy harvesting will facilitate the construction of handheld battery-free devices. Nevertheless, practicality remains a distant prospect, given that modern gaming devices demand significantly more energy than an 8-bit Game Boy. ENGAGE has taken an initial step towards deeply interactive intermittent systems, paving the way for further research. Particularly, the newly designed checkpointing system appears interesting and promising.

REFERENCES

[1] ZF Friedrichshafen AG. Energy harvesting generator. 2015.

[2] Saad Ahmed, Muhammad Hamad Alizai, Junaid Haroon Siddiqui, Naveed Anwar Bhatti, and Luca Mottola. Poster abstract: Towards smaller checkpoints for better intermittent computing. *Proc. IPSN (Nov. 11-13). ACM/IEEE, Porto, Portugal*, 2018.

[3] Saad Ahmed, Muhammad Hamad Alizai, Junaid Haroon Siddiqui, Naveed Anwar Bhatti, and Luca Mottola. Efficient intermittent computing with differential checkpointing. page 70–81, 2019.

[4] Simon Dühne, Johannes Betz, Martin Kolek, Richard Schmuch, Martin Winter, and Tobias Placke. Toward green battery cells: Perspective on materials and technologies. *Small Methods*, 4(7):2000039, 2020.

[5] Ravi Jain and John Wullert. Challenges: Environmental design for pervasive computing systems. page 263–270, 2002.

[6] Alireza Khaligh, Peng Zeng, and Cong Zheng. Kinetic energy harvesting using piezoelectric and electromagnetic technologies—state of the art. *IEEE Transactions on Industrial Electronics*, 57(3):850–860, 2010.

[7] Wojciech Mrozik, Mohammad Ali Rajaeifar, Oliver Heidrich, and Paul Christensen. Environmental impacts, pollution sources and pathways of spent lithium-ion batteries. *Energy Environmental Science*, 14(12):6099–6121, 2021.

[8] Benjamin Ransford, Jacob Sorber, and Kevin Fu. Mementos: System support for long-running computation on rfid-scale devices. *SIGARCH Comput. Archit. News*, 46(3):159–170, 2011.

[9] Jordan Sirani. The best-selling video game consoles of all time. 2023. Last Accessed: 19.12.2023.

[10] Nguyen Van Toan, Truong Thi Kim Tuoi, Nguyen Van Hieu, and Takahito Ono. Thermoelectric generator with a high integration density for portable and wearable self-powered electronic devices. *Energy Conversion and Management*, 245:114571, 2021.

[11] Hui Wang, Peitao Zheng, Huan Yi, Yinyan Wang, Zhuohong Yang, Zhiwen Lei, Yukun Chen, Yonghong Deng, Chaoyang Wang, and Yu Yang. Low-cost and environmentally friendly biopolymer binders for li–s batteries. *Macromolecules*, 53(19):8539–8547, 2020.

[12] Jungang Zhang, Rupam Das, Nosrat Mirzai Jinwei Zhao, John Mercer, and Hadi Heidari. Battery-free and wireless technologies for cardiovascular implantable medical devices. *Advanced Materials Technologies*, 7(6):2101086, 2022.