International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

Design and Implementation of Whack a Mole Game using VHDL

Vaibhavi Sawant[#1], Sarthak Dhumal[#2],Utkarsha Sathe[#3],Unnati Phulmante[#4,] Abhishek Sutar[#5],Rajveer [#6]

#Department of Electronics and Telecommunication Engineering
Vidya Pratishthan's Kamalanayan Bajaj Institute of Engineering and Technology
Savitribai Phule University, Pune, India

Date of Submission: 10-06-2023 Date of Acceptance: 20-06-2023

ABSTRACT–
This research paper presents the design and implementation of a Whack-A-Mole game using VHDL (Very High-Speed Integrated Circuit Hardware Description Language). The objective is to create a digital version of the popular arcade game by utilizing hardware description techniques. The game is implemented on a FPGA (Field-Programmable Gate Array) platform and involves designing the game logic, user interface, and interaction between the hardware components. The results show successful implementation of the game with accurate functionality and user experience.
Keywords - VHDL, Whack-A-Mole game, FPGA,

I. INTRODUCTION:

The purpose of this project is to develop a digital version of the Whack-A-Mole game using VHDL. The game will be implemented on the Basys 3 development board, providing a platform for hardware description and simulation. The objective of the game is to hit randomly appearing moles with a mallet, mimicking the gameplay of the classic arcade game.

Whack-a-Mole game. In this game, you have a grid ofmoles randomly pop up from different moles. The objective is to hit the moles asquickly as possible by pressing corresponding buttons.

Here's a brief overview of how the Whack-a-Mole game typically works:

The game consists of a board with several moles where moles can appear. Each switch underneath it to detect when a mole is whacked. The board may also feature a display to show the player's score and a timer indicating the remaining time.

The concept of Whack-a-Mole:

The concept of "Whack-a-Mole" is a popular arcade game where players use a mallet or a similar tool to hit toy moles that randomly pop up from different holes in a game board. The objective is to hit as many moles as possible within a given time limit.On the Basys 3 board, you can create a digital version of the Whack-a-Mole game using its programmable logic and integrated components.

Whack a mole block diagram as shown in fig.

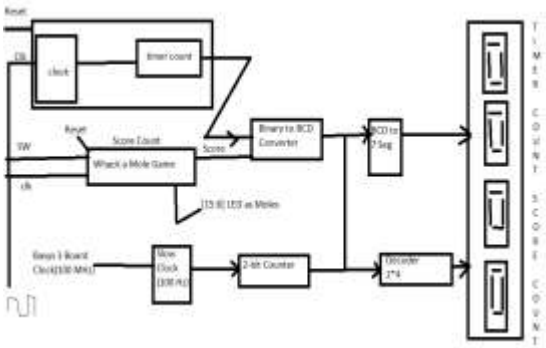

Fig.1 Block diagram of Whack a mole game

Whack-a-Mole games often have multiple difficulty levels, with faster mole appearances or additional challenges as the player progresses. These variations add excitement and increase the level of skill required to achieve high scores.

In arcade settings, Whack-a-Mole games often feature multiple player stations side by side, allowing players to compete against each other simultaneously. Players strive to achieve the highest score within the time limit, fostering a competitive and engaging experience.

To create a Whack a mole game on Basys 3 board we need to create a random generation of numbers on LEDs or seven-segment displays on the Basys 3 board. Generated random number represents the random lightning of LEDs. Here buttons or switches acts as whacks.

We can also Include additional hardware components such as a score counter. We can use a random number generator module to generate a

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

random value after that depending on the generated value, we can activate the corresponding LED or seven-segment display. By controlling the timing and duration of the random number generation we can keep track of the game state, such as the current score. To implement it on Basys 3 we need to create an entity in VHDL and define the inputs, outputs, and internal signals of the game module.

Whack-a-Mole is a simple yet entertaining game that tests players reflexes and hand-eye coordination. It is commonly found in amusement parks, arcades, and entertainment centers. The game's fast-paced nature, combined with the anticipation of moles popping up unpredictably, makes it a fun and challenging experience for players of all ages.[1].

FPGA Board (BASYS 3):

The Basys 3is a popular development board designed for teaching and learning digital design and FPGA (Field-Programmable Gate Array) development. It is manufactured by Diligent, which is known for producing high-quality and affordable FPGA development boards. The Basys 3 board offers a wide range of features and capabilities, making it suitable for various educational and prototyping projects.

Key Features of the Basys 3 Board:

FPGA: The board features a Xilinx Artix-7 FPGA, which is a mid-range FPGA offering a good balance between performance and cost. The Artix-7 FPGA provides a high level of programmable logic cells, memory blocks, and DSP slices, allowing users to implement complex digital designs.

I/O Interfaces: The Basys 3 board offers a variety of I/O interfaces, including:16 user switches: These switches can be used as inputs to the FPGA for different purposes, such as user inputs or configuration options [2].

16 user LEDs: These LEDs can be controlled by the FPGA to provide visual feedback or indicators.

5 push-buttons: The buttons provide additional user inputs for interaction with the FPGA design.

4-digit seven-segment display: This display can be used to show numeric values or other alphanumeric characters.

VGA port: The board has a VGA port for connecting a monitor or display, allowing users to create graphical output from their FPGA designs.

USB-UART bridge: The USB-UART bridge enables communication between the board and a computer, making it easy to program the FPGA and exchange data.

PMOD connectors: The Basys 3 board has four PMOD connectors, which are standardized expansion ports for connecting various peripheral modules, such as sensors, communication modules, or motor controllers.

Programming and Configuration: The Basys 3 board can be programmed and configured using the Xilinx Vivado Design Suite. The suite provides a comprehensive development environment for FPGA designs, including synthesis, simulation, implementation, and programming tools. The board can be programmed using a USB cable connected to the computer [4].

The Basys 3 board can be powered either through the USB connection or an external power supply. It supports a wide input voltage range to accommodate various power sources.

VIVADOSOFTWARE:

Vivado Design Suite is a software tool developed by Xilinx, which is used for designing, implementing, and analysing digital systems using hardware description languages (HDLs) like VHDL. Vivado supports FPGA (Field-Programmable Gate Array) and SoC (System-on-Chip) designs, providing a comprehensive platform for hardware development [4].

Here is an explanation of the key features and functionalities of Vivado Design Suite:

Design Entry: Vivado supports various methods for entering and designing your digital system. You can use the Vivado Integrated Design Environment (IDE) to create or import your design files written in VHDL. The IDE offers a user-friendlygraphical interface where you can design and manage your project.

Synthesis: Once you have entered your design, Vivado' s synthesis tool analyzes your VHDL code and converts it into a gate-level representation. It optimizes the design for area, performance, and power, generating a netlist that represents the internal structure of your digital system.

Implementation: In the implementation stage, Vivado maps the synthesized netlist onto the target FPGA or SoC device. It performs technology mapping, placement, and routing, assigning logic elements and interconnections to achieve the desired functionality. Vivado also handles timing constraints and optimizations to ensure proper operation of the design.

Simulation: Vivado provides a built-in simulator that allows you to verify and validate your VHDL design before synthesis and implementation. You can create testbenches to simulate the behavior of your design and check for correctness and functional accuracy. The simulator supports advanced debugging features for troubleshooting and waveform analysis.

Verification: Vivado offers various verification features to ensure the correctness of your design. It supports formal verification techniques, such as static

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

timing analysis, to check for violations of timing constraints. You can also use the Vivado Integrated Logic Analyzer (ILA) to capture and analyze internal signals within your design for debugging purposes.
Debugging and Analysis: Vivado provides a range of debugging and analysis tools to help you identify and resolve issues in your VHDL design. It offers visual feedback through waveforms, timing diagrams, and resource utilization reports. You can also use the VivadoTclConsole and Vivado Lab Edition for in-system debugging on hardware.
IP Integration: Vivado includes a vast library of pre-built IP (Intellectual Property) cores that you can integrate into your VHDL design. These IP cores provide ready-to-use functions and peripherals, saving design time and effort. Vivado also supports creating custom IP cores that can be reused in future designs [8].
Design Constraints: Vivado allows you to specify design constraints using the Constraints Language (XDC), which helps in controlling the physical implementation and performance of your design. You can define constraints related to timing, placement, clocking, I/O, and other aspects to meet your design requirements.
System Integration: With Vivado, you can integrate multiple subsystems and IPs into a complete system design.It supports hierarchical design methodologies, allowing you to divide your design into manageable blocks and create interconnections between them. Vivado helps with system-level validation and integration.
Device Configuration and Programming: Once your design is complete, Vivado assists in configuring the target FPGA or programming the SoC device. It generates bitstream files that contain the configuration data for the device, which can be loaded onto the hardware for execution.

Random Number Generator:

A random number generator (RNG) is a computational or physical device that generates a sequence of numbers that lacks any discernible pattern or predictability. The purpose of an RNG is to provide a source of randomness or unpredictability in various applications, ranging from computer simulations and cryptography to games and statistical sampling.

There are two main types of random number generators:
Pseudo-Random Number Generators (PRNGs): PRNGs are algorithms that use mathematical formulas or algorithms to generate a sequence of numbers that appear random. However, the generated sequence is deterministic, meaning that if you know the initial seed value, you can reproduce the exact sequence of numbers. PRNGs typically rely on a starting seed value and generate subsequent numbers based on the seed using mathematical operations. The sequence appears random due to the complexity of the algorithms and the use of feedback mechanisms. Common PRNG algorithms include linear congruential generators (LCGs) and Mersenne Twister.

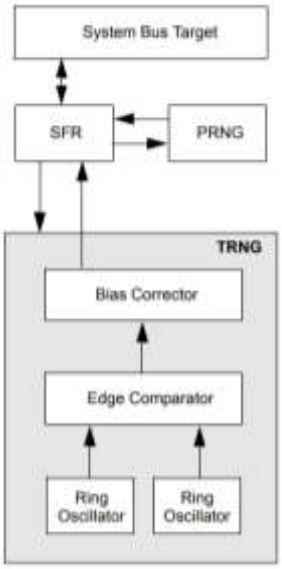

Fig. 2: Random Number Generator Block Diagram [3].

True Random Number Generators (TRNGs): TRNGs generate random numbers based on inherently unpredictable physical processes rather than algorithms. They rely on physical phenomena, such as atmospheric noise, radioactive decay, thermal noise, or electronic circuitry noise, to generate random values. TRNGs capture these random signals and convert them into a digital format. Since they rely on natural sources of randomness, TRNGs are considered more truly random than PRNGs. However, TRNGs can be more expensive and require specialized hardware components.

When selecting an RNG, it's important to consider the specific requirements of the application. For many applications, such as simulations or games, a good-quality PRNG is often sufficient. On the other hand, security-related applications, such as cryptography or secure key generation, typically require the use of TRNGs to ensure the highest level of randomness and unpredictability.

It's worth noting that while random number generators can produce sequences that appear random, they are still generated using deterministic algorithms or physical processes. Therefore, they are technically "pseudo-random" or "true random" within the context of the methods used to generate them.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

Overall, random number generators are essential tools in various fields that require random or unpredictable data. The choice between PRNGs and TRNGs depends on the level of randomness required and the specific application's needs.

Logic for random no.generator:Start with a seed value: The seed value is an initial value used to start the random number generation process. It can be any number, but it should be different for each execution to ensure different sequences of random numbers. You can use the current timestamp or a combination of other variables like system time, process ID, or user input to set the seed value.

Set up a formula or algorithm: Choose a formula or algorithm that generates a sequence of seemingly random numbers based on the seed value. One common algorithm is the linear congruential generator (LCG). The LCG generates numbers using the formula: **Xn = (a * Xn-1 + c) mod m**, where **Xn** is the current random number, **Xn-1** is the previous random number, **a**, **c**, and **m** are constants.

Generate the random number: Apply the chosen formula or algorithm to generate the next random number based on the current seed value. Update the seed value with the newly generated random number.

Repeat step 3 as many times as needed to generate the desired number of random numbers.

FPGA design:

FPGA (Field-Programmable Gate Array) design involves creating digital logic circuits or systems using programmable hardware devices. Unlike traditional integrated circuits, FPGAs can be reprogrammed or reconfigured to implement different functions or designs after manufacturing. This flexibility makes FPGAs popular in various fields, including digital signal processing, embedded systems, telecommunications, and high-performance computing.

Artyxfamily:

The logical layout of a A configurable logic block (CLB) can be seen in the Figure below.

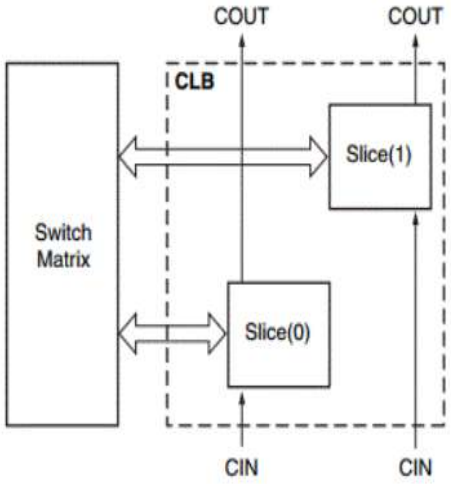

Fig. 3: Logic block diagram of Artix-7

FPGA Architecture and design flow: FPGAs consist of a matrix of programmable logic blocks (PLBs) interconnected by a network of programmable routing resources. Each PLB contains lookup tables (LUTs) that can implement combinational logic functions, as well as flip-flops or registers for sequential logic. The routing resources allow signals to be routed between different logic blocks.

The Artyx family of FPGA is a series of development boards from Diligent that use the Xilinx Artix-7 FPGA chips. These boards are designed with versatility and flexibility in mind, and they have Arduino™ headers and multiple Pmod™ ports for easy expansion. The Artix-7 FPGA chips are part of the Xilinx 7 series of FPGAs, which offer high performance, low power consumption, and high integration[2]. The Artix-7 FPGA chips have up to 215K logic cells, up to 16 x 6.6G transceivers, up to 13Mb of block RAM, and support for DDR3-1066 memory. The Artyx family of FPGA boards can be used for various applications, such as software-defined radio, embedded vision, robotics, and IoT []

Field Programmable Gate Arrays (FPGA) are the semiconductors which are connected using programmable interconnectors. Therefore, the block diagram of FPGA consists of censors,controllers,memories, etc. Controllers used in FPGA are Sensor

Design Entry: FPGA designs can be entered using hardware description languages (HDLs) such as VHDL (VHSIC Hardware Description Language) or Verilog. HDLs provide a textual representation of the desired digital circuit or system, allowing designers to describe the behaviour and structure of the design [7].

Synthesis: After the design is entered in an HDL, synthesis tools are used to convert the high-level description into a gate-level representation.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

Synthesis involves mapping the HDL code to specific FPGA resources, such as LUTs and flip-flops, and optimizing the design for performance, area, or power consumption [5].

Place and Route: The place-and-route (P&R) process determines the physical locations of the design's logic elements within the FPGA and establishes the interconnections between them. P&R tools take into account the constraints provided by the designer, such as performance goals, pin assignments, and resource utilization, to generate an optimized layout and routing solution.

Timing Analysis: Timing analysis is performed to ensure that the design meets the required timing constraints. It checks the delays through the circuit and verifies that all signal paths can meet the specified performance requirements. Timing analysis helps identify and address potential timing violations, such as setup and hold time violations.

Simulation and Verification: Simulation is an essential step in FPGA design to verify the functionality and correctness of the design before synthesis and implementation. Simulation tools simulate the behaviour of the design using test vectors or stimulus, allowing designers to validate the design's functionality and identify potential issues or bugs.

Implementation: Once the design is synthesized, the synthesized netlist is used to program or configure the FPGA device. The configuration bitstream, generated during the implementation process, contains the information needed to configure the FPGA with the desired logic functionality.

Debugging and Testing: After implementation, designers perform debugging and testing to ensure that the FPGA design operates correctly. This involves analysing the behaviour of the design, probing signals, and using debugging tools to identify and resolve any functional or performance issues.

FPGA Development Tools: FPGA development is supported by vendor-specific design tools, such as Xilinx Vivado, Intel Quartus Prime, or Lattice Diamond. These tools provide a complete design environment, including synthesis, simulation, implementation, and programming capabilities.

BCD TO 7 SEGMENTS:

BCD (Binary Coded Decimal) is a numerical representation system that uses a combination of four binary digits (bits) to represent each decimal digit from 0 to 9. On the other hand, a 7-segment display is a common type of electronic display device that can show the digits 0 to 9 and some additional characters using seven individual segments.

Decimal	Input Line				Output Line							Display
Digit	A	B	C	D	a	b	c	d	e	f	g	Pattern
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	1	0	1	1	9

Fig.4: Result of BCD to Seven segment

To convert BCD to a 7-segment display format, you need a BCD to 7-segment decoder. This decoder is a combinational logic circuit that takes a BCD input and produces the appropriate combination of signals to illuminate the corresponding segments of the 7-segment display.

A typical BCD to 7-segment decoder has four inputs, labelled A, B, C, and D, representing the four BCD bits. These inputs can have binary values from 0000 to 1001, which correspond to decimal digits 0 to 9. The decoder also has seven outputs, labelled a, b, c, d, e, f, and g, which are connected to the individual segments of the 7-segment display.

Each output of the decoder is associated with a particular segment of the display. The segments are usually labelled a through g, where a, b, c, d, e, f, and g represent the segments that form the numerical digits.

To convert a BCD digit to a 7-segment format, you map the input BCD bits to the appropriate outputs of the decoder. For example, if the BCD input is 0000, the decoder will activate the segments corresponding to the digit 0. If the BCD input is 0101, the decoder will activate the segments corresponding to the digit 5.

The specific mapping between BCD inputs and 7-segment outputs depends on the implementation of the decoder. Different decoders may have slightly different arrangements of the output signals. However, in most cases, the mapping is designed to provide a logical and intuitive representation of the decimal digits on the 7-segment display.

II. RESULTS AND CONCLUSION:

Whack-a-Mole is a classic arcade game that has captivated players for decades. It’s simple yet addictive gameplay, where players must react quickly to hit moles as they randomly appear, provides an enjoyable and challenging experience. The game's emphasis on reflexes and hand-eye coordination makes it engaging for players of all ages.

Whack-a-Mole's mechanics are straightforward: players use a mallet or striking tool to

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

hit the moles as they pop up from their holes. Successful hits earn points, and the game typically includes a timer to add a sense of urgency. Multiple difficulty levels can be implemented to increase the challenge and keep players engaged.

The game's competitive nature is enhanced in arcade settings, where multiple players can compete side by side to achieve the highest score within a time limit. This adds an element of excitement and encourages players to improve their skills.

Whack-a-Mole's enduring popularity is a testament to its simplicity and the thrill it provides. Whether played in arcades, amusement parks, or even as digital versions on mobile devices, the game continues to entertain and test players' abilities to react swiftly and accurately.

Overall, Whack-a-Mole remains a beloved game that combines quick reflexes, hand-eye coordination, and competitive spirit, making it a timeless classic in the world of arcade gaming.

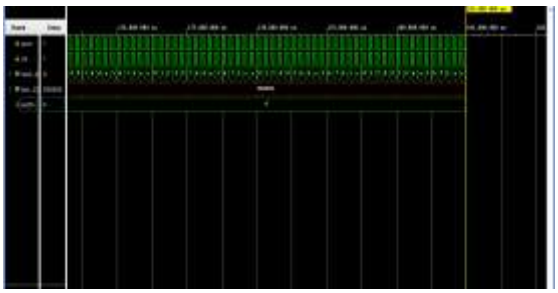

Fig. 5: Simulation Results of random number generator

The result for the random number generator which is created by using the xor gate logic is shown here.To get the required random number we need to give clock and push button as input in simulation.Hence here we gave the rising edge of clock as 1 and trailing edge of clock as 0, The force constant for push button is given as Every time after the push button is clicked the new random number is generated at the output of simulation.

For the random number generation, we have to set the push button at 1(high). The width is taken as 4 HereWe can set the limit of random number generation also.

Fig.6: Simulation Results of BCD to Seven Segment

BCD (Binary Coded Decimal) to Seven segment decoder is used to displayBCDinput as output on 7 segment display.4bitBCD number is given as the input to the decoder and the information is displayed on seven individual colored LED's (called the segments).

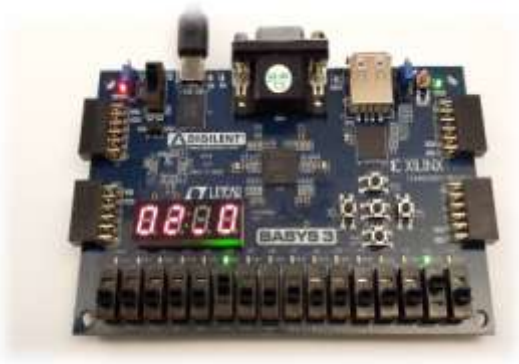

Fig.7: Output of Whack a mole game.

In a Whack-a-Mole game, the main objective is to hit or "whack" as many moles as possible within a given time limit. The game is usually played on a console or a digital platform.

Synthesis Report:

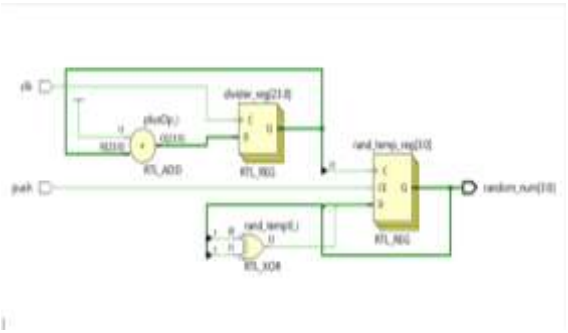

Fig.8: Digital diagram of BASYS 3

A digital diagram of BASYS 3 is a schematic representation of theBasys 3 board, which is a complete, ready-to-use digital circuit development platform based on the latestArtix-7™ Field Programmable Gate Array (FPGA) from Xilinx. The diagram shows the various ports and peripherals that are available on the board, such as switches, LEDs, buttons, 7-segment display, VGA, USB, Pmod connectors, etc. The diagram also shows the power supply circuit and the FPGA configuration circuit[1]. The digital diagram of BASYS 3 can help you understand how to connect and use the board for different designs and applications.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 5, Issue 6 June 2023, pp: 478-484 www.ijaem.net ISSN: 2395-5252

Power:Power analysis given below shows the Total On-Chip power which is given as 4.132 watt. The total power distribution on the project is also shown here. Signals consume 0.132-watt power, logic takes 0.046-watt power, I/O consumes 96% of power i.e., 3.766 watt. Power analysis diagramcan be seen in the Figure below.

Fig.9: Power analysis of Whack a mole game

Timer:Timing summary below shows the total number of endpoints for the project implementation i.e.,38. Time summary diagramcan be seen in the Figure below.

Fig.10: Timing summary of Whack a mole game

REFERENCE:

[1]. N. Ascheuer, S. O. Krumke, and J. Rambau. Online dial-a-ride problems: Minimiz-ing the completion time. In Proceedings of the 17th International Symposium on

[2]. Theoretical Aspects of Computer Science, volume 1770 of Lecture Notes in Com-puter Science, pages 639–650. Springer, 2000.

[3]. G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, and M. Talamo. Algorithmsfor the on-line traveling salesman. Algorithmica, 29(4):560–581, 2001.

[4]. S. Baruah, J. Haritsa, and N. Sharma. On-line scheduling to maximize task comple-tions. The Journal of Combinatorial Mathematics and Combinatorial Computing,39:65–78, 2001.

[5]. M. Blom, S. O. Krumke, W. E. de Paepe, and L. Stougie. The online-TSP againstfair adversaries. Informs Journal on Computing, 13(2):138–148, 2001.

[6]. A. Borodin and R. El-Yaniv. Online Computation and Competitive Analysis. Cam-bridge University Press, 1998.

[7]. W. E. de Paepe, J. K. Lenstra, J. Sgall, R. A. Sitters, and L. Stougie. Computer-aided complexity classification of dial-a-ride problems. Informs Journal on Com-puting, 2003. To appear.

[8]. E. Feuerstein and L. Stougie. On-line single server dial-a-ride problems. TheoreticalComputer Science, 268(1):91–105, 2001.

[9]. M. R. Garey and D. S. Johnson. Computers and Intractability (A guide to thetheory of NP-completeness). W.H. Freeman and Company, New York, 1979.

[10]. D. Hauptmeier, S. O. Krumke, and J. Rambau. The online dial-a-ride problemunder reasonable load. In Proceedings of the 4th Italian Conference on Algorithmsand Complexity, volume 1767 of Lecture Notes in Computer Science, pages 125–136. Springer, 2000.

[11]. S. Irani, X. Lu, and A. Regan. On-line algorithms for the dynamic traveling repairproblem. In Proceedings of the 13th Annual ACM-SIAM Symposium on DiscreteAlgorithms, pages 517–524, 2002.

[12]. B. Kalyanasundaram and K. R. Pruhs. Maximizing job completions online. InProceedings of the 6th Annual European Symposium on Algorithms, volume 1461of Lecture Notes in Computer Science, pages 235–246. Springer, 1998.

[13]. R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W.Thatcher, editors, Complexity of Computer Computations, pages 85–103. PlenumPress, New York, 1972.