Faculty of Engineering Sciences

Heidelberg University

Master's Thesis

in Computer Engineering

submitted by

Bc. VLADISLAV VÁLEK

born in Šumperk, Czech Republic

2026

SHARED CPU FPGA FILESYSTEM ACCESS ON FAST NVME DEVICES

This Master's Thesis has been carried out by

Bc. Vladislav Válek

at the

Institute of Computer Engineering (ZITI)

under the supervision of

Prof. Dr. Dirk Koch

ABSTRACT

Abstract in English.

KEYWORDS

PCIe, PCI Express, DMA, Direct Memory Access, host-bypassing, peer-to-peer, SPDK, NVMe, FPGA

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg ....

Unterschrift

ACKNOWLEDGEMENT

I would like to thank my supervisor, Prof. Dr. Dirk Koch for his valuable feedback when writing this thesis as well as in my research work at the Heidelberg University. I would also like to thank my colleague, Dr. Riadh Ben Abdelhamid, for providing necessary user feedback for the current course of research. Further, I give many thanks to my colleagues at CESNET, a.l.e., Ing. Jakub Cabal and Ing. Martin Špinler, for mentoring me in the development of PCIe-compliant IPs and software development. Finally, I thank my mom, Ing. Marta Válková, for supporting me throughout my studies.

Contents

List of Figures

List of Tables

List of Algorithms

List of Listings

Introduction

Here comes the introduction of the thesis, for example...

This thesis is devoted to **DSP!** (**DSP!**), especially it analyses the effect happening when the Nyquist condition for ***symfs!*** (**symfs!**) is not satisfied.[1]

The template is set to twoside printing by default. Do not be surprised that you find empty pages in your PDF. They are there to make the chapters and other important stuff begin on the right side when the document is printed. Having a serious reason to print one-sided, please switch the option `twoside` to `oneside`!

[1] This sentence is only to demonstrate how abbreviations can be used and typeset.

1 PCI Express

The *Peripheral Component Interconnect Express* (PCIe) is the interconnect standard that emerged in 2003 from the *Peripheral Component Interconnect* (PCI) standard. The PCIe addressed the drawbacks of the PCI that became increasingly challenging when more devices and more throughput were required. These include:

- Shared-bus topology that, inspite of its simplicity, was rendered unscalable since with increasing amount of devices, the bus gets longer which weakens the signal for the farthest device.
- Common clock which does not fit well with parallel data transmission model especially when lane skew takes place, meaning the bits of data travel unevenly fast over the wires.
- Half-duplex communication allows only one device to communicate at one point at a time.
- Reflected-wave signaling which was used to reduce the power consumption on the bus. This uses an interference of the sent wave from the master (which had driven the signal to the lower voltage than needed for the digital circuits on the bus) and its reflection that originated on the end of a bus (there is no termination on the bus' end). By the addition of these two waves, the signal reaches the required voltage level for the digital circuits (W: cite PCIe book).

The PCI protocol did not keep pace with increasing demand of computer peripheral communication structures and needed to be changed. The PCI Express protocol significantly changed the strategy for both the communication topology and the physical properties of the interconnect.

1.1 Communication model

The basic architecture is shown in Fig. XXX (host topology). Connections are implemented as point-to-point links between ports on different device types organized in a tree. The center of the topology is the *Root Complex* (RC), which provides *Root Ports*. Each device communicating over the infrastructure (such as today's GPU/F-PGA accelerators) is called *Function* and is the major source of traffic within the system[1]. Each Function contains an *Upstream port* for exchanging data. Since the number of Root Ports is limited by the processor's interconnect, fan-out is provided by *PCIe Switches*, which supply multiple *Downstream ports* and a single Upstream port. Each port supports full-duplex communication and introduces asynchronous clocking model. Using this model significantly improves the possible throughput of the interconnect where clock gets reconstructed from the patterns in the incoming data on the receiving port. Using Switches, the number of devices can be increased

arbitrarily, making PCIe a switched network (W: cite PCIe spec). Although the PCIe specification introduces many advanced features, this thesis presents mostly the necessary ones for its purposes.

Devices communicate using units called *Transaction Layer Packets* (TLPs), forming a packetized transfer. Each TLP contains a PCIe header used for identification followed by an optional payload. There are many TLP types, but the ones mentioned in this work are *Memory Read Request* (MRd) and *Memory Write Request* (MWr). Requests may require a response (called *non-posted*, as with MRds) in the form of *Completion* (Cpl) presenting a third TLP type. Such requests are called *non-posted* (like MRds) while requests that do not require a response are called *posted* (like MWrs). The device creating a request is called *Requester*, while the device 'completing' the request (sending a response or just accepting the payload) is called *Completer*.

The RC is located on the CPU die and tightly integrated with the processing cores to provide fast access to the system's I/O devices. The RC also provides the connection to host memory, which is the primary communication path used by DMA engines for *Host-to-Device* (H2D) and *Device-to-Host* (D2H) transfers. Multiple Root Ports or Downstream ports enable direct *Device-to-Device* (D2D) communication, as shown in Fig. XXX.

1.2 Link configuration

The PCIe specification gets published in versions called *generations*. With every new specification, the throughput of the bus gets doubled while keeping backward and forward compatibility between its standards (this is an important feature of PCIe). This is the reason, why the physical implementation of the protocol stack remains so complicated, especially on the physical layer. The physical connection between two ports consists of multiple differential pairs called *lanes* that are bundled together into a *link*. Each lane allows for full-duplex communication, where each direction is communicated using a separate differential pair. The link can either contain 1, 2, 4, 8 or 16 lanes.

As for the physical attachment interface, this thesis works with two of them, namely *Card Electromechanical* (CEM) and *M.2*. The CEM standard is the oldest

[1]The PCIe specification (with concordance to the PCI) defines three *System elements* to identify its devices, namely *Bus*, *Device* and *Function*. Every point-to-point connection in the topology makes a separate Bus together with internal (virtual) buses of the Switches and the Root Complex. The Device identifier stays as a remnant of the PCI standard and every Bus within the PCIe topology contains only one. Within each Device, one or multiple Functions can be present. The Function is basically and adressable entity in the configuration space where *Endpoint* is the most important type of a Function. Using the term 'device' in this thesis will refer to a general device in the PCIe topology that can contain one or multiple Functions (or Endpoints).

one used for *Add-in Cards* with GPU chips, FPGA accelerators or *Network Interface Cards* (NICs). It supports from 1 to 16 lanes with with wider link widths supporting lower ones and vice versa (these however with specially carved connectors that fit longer connector of the attached card). This form-factor allows for connection of the most power consuming peripherals where the connector can supply up to 75 W. The accelerators can require additional power by introducing external power supply attachements. The most common ATX connector adds another 75 W for its 6-pin configuration or 150 W for the 8-pin configuration (W: cite Intel's ATX standard).

The second type of physical attachement, *M.2*, is designed for ultra-light, power efficient platforms such as wireless modules or (as in the case of this thesis) *Non-volatile Memory Express* (NVMe) drives. This connector has a fixed set of 4 lanes and a variable module length ranging from 30 mm to 110 mm. These connectors present a detachable form factor for PCIe links but the standard allows for permanently attached devices to be connected as well.

As it is obvious from the variety of link configurations in terms of speeds and the amount of lanes, the multitude of PCIe devices inside a compute node needs to agree on common parameters in order to facilitate data transfer. This process of negotiation is called *link training* and it takes place after the network get powered up. The training is done on per-link-basis so there are multiple differently configured links in the system at a given point in time. This is presented on Fig. XXX where the M.2 SSD needs to communicate with the processor (where Root Ports can usually utilize up to 16 lanes) or with a NIC which supports up to 16 lanes. Other than link's lane count and transfer rate, other configured parameters include lane polarity and lane reversal. Since the training and the negotiated link parameters are tha matter of the physical layer, the link properties remain transparent for the user.

1.3 Device configuration

After the initial link training is done and every link is active, the host system does device discovery and configuration upon system boot in order to map the network. The process of discovery is called *enumeration* and involves a CPU communicating through the Root Complex in order to find every connected device in the topology and assign its BDF identifier (this includes assigning numbers also to buses between ports that do not necessarily connect to an Endpoint, like two Switches or a Switch with the Root Complex). The search is done in a depth-first way where each device is registered if it responds with a valid *Vendor ID* that is located within device's configuration registers. After assigning all BDFs in the topology, the configuration software further configures registers on the discovered devices. The configuration is done solely by the Root Complex in order to avoid the complexity of management

when multiple devices (e.g. Endpoints) would try to configure each other. The initial assigning of configuration attributes is done using an IO-based transfer but further setting is done using *Memory-mapped I/O* (MMIO).

Each Endpoint contains configuration registers in a 64-byte *PCI Configuration Header*, which holds the most important attributes controlling the Function's ability to initiate transactions over PCIe and to be addressable within a host system. Firstly, since devices cannot generate Requests on their own by default, the *Bus Master* attribute enables this feature. Secondly, in order for adjacent devices to locate the current Function and to access its internal, user-defined registers/memory space, at least one of the *Base Address Registers* (BARs) must be initialized. By using BARs, each Endpoint requests a memory space within a host system for MMIO access. This is needed to access the application logic from the outside since each Endpoint accepts only those transactions whose addresses are in the ranges specified by its BARs. Each Function can register up to six 32-bit BARs which provides memory segmentation needed by some host applications. When 64-bit addressing is requested, two consecutive BARs are used. Throughout this work, the observation has been made that most devices allocate one to two BARs and, when not exceeding 4 GiB of space, an operating system maps them to 32-bit address ranges even when 64-bit addressing has been requested. First of all, this addressing has to be enabled in the UEFI settings of the host system.

Besides than the sheer capability for devices to be localizable in the system, other parameters are configured and need to be abided by by the user applications (e.g. the user-defined FPGA logic). A function defines its supported *Maximum Payload Size* (MPS) in the *Device Capabilities register*. Seeing this value, the configuration software sets its preferred value in a field in the *Device Control regiser*. The value of this field prohibits the Transmitter to dispatch TLP with greater payload than MPS specified and the Receiver to process them. The configuration allows furter adjustement using the *Maximum Read Request Size* (MRRS) in the *Device Control register* which prohibits the Transmitter to dispatch a Read Request greater than the specified size. Available values for these two parameters range from 128 to 4096 bytes (in powers of two steps). The configuration software balances these parameters based on latency/bandwidth requirements in the network. The configuration has to take account of the pairs of adjacent devices communicating with each other since receiving a TLP on a function's port whose size is larger than its set MPS results in the function rejecting this packet and reporting a fatal error[2].

[2]The PCIe standard does not define a segmentation of large TLPs.

1.4 Peer-to-peer transfer

An important aspect of PCIe (with its predecessor as well) and the structure of its network allowed to introduce *Direct Memory Access* (DMA) that allows to copy large amounts of data withouth the host CPU facilitating such copying. The usual approach it the exchange of data between PCIe devices and the host CPU through buffers in the host memory. However, the network infrastructure to address other devices in the topology as well. This type of D2D transport is called *Peer-to-peer* (P2P) or *Host Bypassing* and makes a fundamental part of this thesis. There are several implications of running such transport in the host system. The specification makes the support for each Switch mandatory (for transport between its Downstream ports) but optional for the Root Complex (for transport between its Root Ports) (source to spec). The routing of TLP differes based on its type with MWrs and MRds being routed by memory address while Cpl being routed by Requester's BDF identifier.

Today's host CPUs untilize *I/O Memory Management Unit* (IOMMU) to translate physical address to virtual ones in order to securely split I/O peripherals into groups. This unit is crucial in multi-tenant environments (like guests as virtual machines) where only devices in one group can perform P2P transfers between each other. Otherwise, where communication between two groups is required, the data need to be analyzed by the CPU. Without IOMMU a potentially malicious device can write and read from every region in the host system. The unit works hand in hand with the PCIe's *Access Control Services* (ACS) which forces every traffic to pass through the RC even if the devices are connected to the single Switch (depicted in Fig XXX). Otherwise, all devices under each Root Port are put to a common IOMMU group which does not play well with the need to decide isolation on per-device basis.

Considering the test server used in this work, the support for P2P transferes has been discovered as well as the presence of multiple IOMMUs and enabled ACS on the Root Ports. Since security is not a topic of this thesis nor are the system considerartions in the multi-tenant environments, both of these funcionalities have been disabled in server's UEFI. Therefore, the PCIe devices communicate using physical addresses.

2 Integrated core for PCIe on FPGAs

3 Thesis Results

Practical part and results of the student, suitably split into chapters and sections.

3.1 Selection of Programming Language

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nulla pulvinar eleifend sem. Integer in sapien. Etiam sapien elit, consequat eget, tristique non, venenatis quis, ante. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Phasellus enim erat, vestibulum vel, aliquam a, posuere eu, velit. Aliquam erat volutpat. Nullam faucibus mi quis velit [1].

3.2 Implementation

Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. Fusce tellus. Morbi scelerisque luctus velit. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Quisque porta. Fusce suscipit libero eget elit. Nulla non lectus sed nisl molestie malesuada. Phasellus faucibus molestie nisl. Integer vulputate sem a nibh rutrum consequat. Proin mattis lacinia justo. Phasellus et lorem id felis nonummy placerat. Etiam ligula pede, sagittis quis, interdum ultricies, scelerisque eu. Cras elementum. Aenean placerat. Donec ipsum massa, ullamcorper in, auctor et, scelerisque sed, est. Aliquam ante. Integer imperdiet lectus quis justo. Vivamus ac leo pretium faucibus. Nullam faucibus mi quis velit.

3.2.1 Tests and Evaluation

Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Aliquam erat volutpat. Lorem ipsum dolor sit amet, consectetuer adipiscing elit [1, 6]. Nunc auctor. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Maecenas lorem. Maecenas libero. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Nullam rhoncus aliquam metus.

Integer rutrum orci vestibulum

Integer rutrum, orci vestibulum ullamcorper ultricies, lacus quam ultricies odio, vitae placerat pede sem sit amet enim. Ut enim ad minim veniam, quis nostrud

exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. Nullam eget nisl. Nunc auctor. Etiam dui sem, fermentum vitae, sagittis id, malesuada in, quam. Fusce dui leo, imperdiet in, aliquam sit amet, feugiat eu, orci. Curabitur vitae diam non enim vestibulum interdum. Aliquam erat volutpat. Pellentesque sapien. Phasellus enim erat, vestibulum vel, aliquam a, posuere eu, velit.

Eger rutrum orci westibulum

Fusce dui leo, imperdiet in, aliquam sit amet, feugiat eu, orci. Maecenas aliquet accumsan leo. Aliquam ornare wisi eu metus. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam erat volutpat. Donec iaculis gravida nulla. Sed elit dui, pellentesque a, faucibus vel, interdum nec, diam. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Nulla non arcu lacinia neque faucibus fringilla. Phasellus enim erat, vestibulum vel, aliquam a, posuere eu, velit. Praesent vitae arcu tempor neque lacinia pretium [7, 8, 9].

Aliquam erat volutpat. Quisque porta. Integer imperdiet lectus quis justo. Nullam justo enim, consectetuer nec, ullamcorper ac, vestibulum in, elit. Nullam faucibus mi quis velit. Fusce tellus. Fusce consectetuer risus a nunc. Cras pede libero, dapibus nec, pretium sit amet, tempor quis. Morbi imperdiet, mauris ac auctor dictum, nisl ligula egestas nulla, et sollicitudin sem purus in lacus [2, 3, 4]. Mauris elementum mauris vitae tortor. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem. Quisque porta. Integer vulputate sem a nibh rutrum consequat. Nulla pulvinar eleifend sem. Praesent id justo in neque elementum ultrices [5].

Fusce suscipit libero eget elit. Integer vulputate sem a nibh rutrum consequat. Aliquam erat volutpat. Etiam neque. Nulla turpis magna, cursus sit amet, suscipit a, interdum id, felis. Nullam rhoncus aliquam metus. Etiam dui sem, fermentum vitae, sagittis id, malesuada in, quam. Nunc auctor. Nunc dapibus tortor vel mi dapibus sollicitudin. Praesent in mauris eu tortor porttitor accumsan. Nulla non arcu lacinia neque faucibus fringilla. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. Maecenas lorem. Aenean placerat. Donec vitae arcu. Maecenas lorem. Donec iaculis gravida nulla. Nulla non lectus sed nisl molestie malesuada.

Duis pulvinar. Nulla est. Duis condimentum augue id magna semper rutrum. Integer pellentesque quam vel velit. Aliquam ante. Nulla quis diam. Proin mattis lacinia justo. Aenean fermentum risus id tortor. Nunc auctor. Nullam justo enim, consectetuer nec, ullamcorper ac, vestibulum in, elit. In dapibus augue non sapien.

Etiam bibendum elit eget erat. In sem justo, commodo ut, suscipit at, pharetra vitae, orci. Maecenas libero.

Nulla non lectus sed nisl molestie malesuada. Donec vitae arcu. Aenean fermentum risus id tortor. Praesent in mauris eu tortor porttitor accumsan. Nulla pulvinar eleifend sem. Duis viverra diam non justo. Integer imperdiet lectus quis justo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. In rutrum. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nulla non lectus sed nisl molestie malesuada. Aliquam erat volutpat. Mauris tincidunt sem sed arcu. Duis bibendum, lectus ut viverra rhoncus, dolor nunc faucibus libero, eget facilisis enim ipsum id lacus. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat. In enim a arcu imperdiet malesuada. Nulla non lectus sed nisl molestie malesuada. Proin mattis lacinia justo.

Aliquam in lorem sit amet leo accumsan lacinia. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Suspendisse sagittis ultrices augue. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. In convallis. Praesent id justo in neque elementum ultrices. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.

Pellentesque pretium lectus id turpis. Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Curabitur ligula sapien, pulvinar a vestibulum quis, facilisis vel sapien. Praesent dapibus. Sed elit dui, pellentesque a, faucibus vel, interdum nec, diam. Duis viverra diam non justo. Duis ante orci, molestie vitae vehicula venenatis, tincidunt ac pede. Phasellus rhoncus. Maecenas fermentum, sem in pharetra pellentesque, velit turpis volutpat ante, in pharetra metus odio a lectus. Proin pede metus, vulputate nec, fermentum fringilla, vehicula vitae, justo. Fusce aliquam vestibulum ipsum. Nullam at arcu a est sollicitudin euismod.

Conclusion

Thesis conclusion.

Bibliography

[1] VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ. *Směrnice č. 72/2017, Úprava, odevzdávání a zveřejňování závěrečných prací.* Online. Brno: VUT v Brně, 2017. Úplné znění ke dni 11. 4. 2022. Dostupné z: `https://www.vut.cz/uredni-deska/vnitrni-predpisy-a-dokumenty/smernice-c-72-2017-uprava-odevzdavani-a-zverejnovani-zaverecnych-praci-d161410`. [cit. 2023-09-27].

[2] ÚŘAD PRO TECHNICKOU NORMALIZACI, METROLOGII A STÁTNÍ ZKUŠEBNICTVÍ. ČSN ISO 690:2022 (01 0197), *Informace a dokumentace – Pravidla pro bibliografické odkazy a citace informačních zdrojů.* Čtvrté vydání. Praha, 2022.

[3] ÚŘAD PRO TECHNICKOU NORMALIZACI, METROLOGII A STÁTNÍ ZKUŠEBNICTVÍ. ČSN ISO 7144 (010161), *Dokumentace – Formální úprava disertací a podobných dokumentů.* Praha, 1997.

[4] ÚŘAD PRO TECHNICKOU NORMALIZACI, METROLOGII A STÁTNÍ ZKUŠEBNICTVÍ. ČSN ISO 31-11, *Veličiny a jednotky – část 11: Matematické znaky a značky používané ve fyzikálních vědách a v technice.* Praha, 1999.

[5] FARKAŠOVÁ, B. et al. *Výklad normy ČSN ISO 690:2022 (01 0197) účinné od 1. 12. 2022.* Online. První vydání. 2023. Dostupné z: `https://www.citace.com/Vyklad-CSN-ISO-690-2022.pdf`. [cit. 2023-09-27].

[6] *Pravidla českého pravopisu.* 1. vydání. Olomouc: FIN, 1998. ISBN 80-86002-40-3.

[7] WALTER, G. G. a SHEN, X. *Wavelets and Other Orthogonal Systems.* 2. vydání, Boca Raton: Chapman & Hall/CRC, 2000. ISBN 1-58488-227-1

[8] SVAČINA, J. Dispersion Characteristics of Multilayered Slotlines – a Simple Approach. *IEEE Transactions on Microwave Theory and Techniques.* 1999, vol. 47, no. 9, s. 1826–1829. ISSN 0018-9480.

[9] RAJMIC, P. a SYSEL, P. Wavelet Spectrum Thresholding Rules. In: *Proceedings of the International Conference Research in Telecommunication Technology.* Žilina: Žilina University, 2002. s. 60–63. ISBN 80-7100-991-1.

Symbols and abbreviations

PCI	Peripheral Component Interconnect
PCIe	Peripheral Component Interconnect Express
DMA	Direct Memory Access
NVMe	Non-volatile Memory Express
FPGA	Field-programmable Gate Array
P2P	Peer-to-peer
RC	Root Complex
EP	Endpoint
TLP	Transaction Layer Packet
DLLP	Data-link Layer Packet
MRd	Memory Read Request
MWr	Memory Write Request
Cpl	Completion
H2D	Host-to-Device
D2H	Device-to-Host
C2H	Card-to-Host
H2C	Host-to-Card
D2D	Device-to-Device
CEM	Card Electromechanical
NIC	Network Interface Card
SSD	Solid-state Drive
MMIO	Memory-mapped I/O
BAR	Base Address Register
MPS	Maximum Payload Size

MRRS	Maximum Read Request Size
IOMMU	I/O Memory Management Unit
ACS	Access Control Services

List of appendices

A Selected Commands of `thesis` Package

A.1 Quantities and Units

Table A.1: An overview of commands (use within the mathematical environments).

Command	Example	LaTeX code of example	Meaning
`\textind{...}`	$\beta_{\max}$	`$\beta_\textind{max}$`	text-style index
`\const{...}`	U_{in}	`$\const{U}_\textind{in}$`	constant
`\var{...}`	u_{in}	`$\var{u}_\textind{in}$`	variable
`\complex{...}`	$\underline{u}_{\mathrm{in}}$	`$\complex{u}_\textind{in}$`	complex variable
`\vect{...}`	$\mathbf{y}$	`$\vect{y}$`	vector
`\mat{...}`	$\mathbf{Z}$	`$\mat{Z}$`	matrix
`\unit{...}`	kV	`$\unit{kV}$` or `\unit{kV}`	unit

A.2 Symbols

- `\E`, `\eul` – typesets the Euler number: e,
- `\J`, `\jmag`, `\I`, `\imag` – imaginary unit: j, i,
- `\dif` – the differential: d,
- `\sinc` – the function sinc,
- `\mikro` – typesets the *micro* symbol in roman type[1]: µ,
- `\uppi` – typesets π (greek pi in roman type, in difference to `\pi`, which typesets π).

All symbols are considered to be used within a math mode, except `\mikro` that is possible in the text mode as well.

[1] the symbol comes from package `textcomp`

B Next Appendix

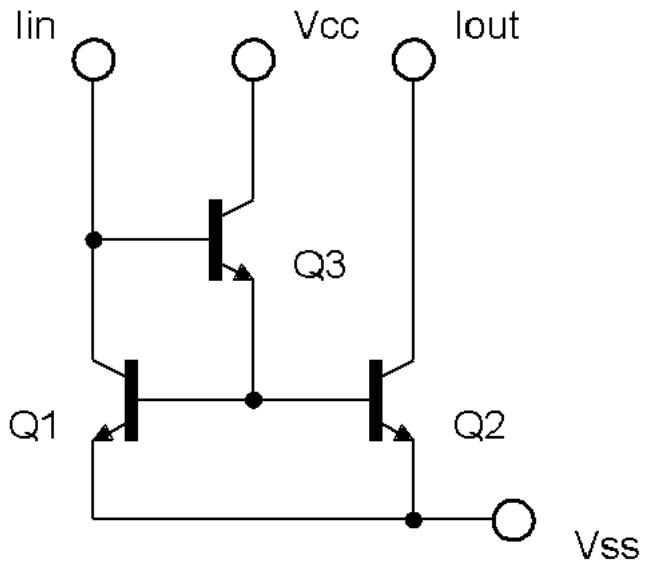

Fig. B.1: Improved Wilson current mirror.

For inclusion of the vector-based graphics directly via LaTeX, it is possible to use the `TikZ` package. Examples of use can be found at the TeXample site. TikZ graphics creation is supported in QTikz and TikzEdt software.

C Examples of Listing Computer Codes

C.1 Package `listings`

Listing computer codes can be handled efficiently via the `listings` package. This package introduces a new environment `lstlisting` for typesetting computer codes, as for example:

```
\section{Package lstlistings}
Listing computer codes can be handled efficiently
via the \texttt{listings} package.
This package introduces a new environment
\texttt{lstlisting} for typesetting computer codes.
```

The package supports a number of programming languages. The code to be typeset can be input directly from files on disk. The package allows row numbering and extracting only selected parts of the code. The following paragraph is an example of the use of `listings`:

Abbreviations are typeset with the `acronym` environment:

```
6 \begin{acronym}[HowMuchSpace]
```

The width of the input parameter, `HowMuchSpace`, determines the width of the first column. An example of the definition of abbreviation **symfs!** is in Listing C.1.

Listing C.1: Example of code listing.

```
102   % \acro{symfs}              % label of the abbrev.
103   %   [\ensuremath{f_\textind{s}}]  % symbol
104   %   {sampling frequency}          % full text
105   %
```

The list is finished with the end of the environment:

```
26     \acro{RC}
```

Listing C.2 contains an example of code for Matlab, whereas in Listing C.3 you find an example in the C language.

Listing C.2: Example of the Schur–Cohn test of stability in Matlab.

```
%% Priklad testovani stability filtru

% koeficienty polynomu ve jmenovateli
a = [ 5, 11.2, 5.44, -0.384, -2.3552, -1.2288];
disp( 'Polynom:'); disp(poly2str( a, 'z'))

disp('Kontrola pomoci korenu polynomu:');
zx = roots( a);
if( all( abs( zx) < 1))
    disp('System je stabilni')
else
    disp('System je nestabilni nebo na mezi stability');
end

disp(' '); disp('Kontrola pomoci Schur-Cohn:');
ma = zeros( length(a)-1,length(a));
ma(1,:) = a/a(1);
for( k = 1:length(a)-2)
    aa = ma(k,1:end-k+1);
    bb = fliplr( aa);
    ma(k+1,1:end-k+1) = (aa-aa(end)*bb)/(1-aa(end)^2);
end

if( all( abs( diag( ma.'))))
    disp('System je stabilni')
else
    disp('System je nestabilni nebo na mezi stability');
end
```

Listing C.3: Example of implementation of first canonical form in C.

```
// first canonical form
short fxdf2t( short coef[][5], short sample)
{
  static int v1[SECTIONS] = {0,0},v2[SECTIONS] = {0,0};
  int x, y, accu;
  short k;

  x = sample;
  for( k = 0; k < SECTIONS; k++){
    accu = v1[k] >> 1;
    y = _sadd( accu, _smpy( coef[k][0], x));
    y = _sshl(y, 1) >> 16;

    accu = v2[k] >> 1;
    accu = _sadd( accu, _smpy( coef[k][1], x));
    accu = _sadd( accu, _smpy( coef[k][2], y));
    v1[k] = _sshl( accu, 1);

    accu = _smpy( coef[k][3], x);
    accu = _sadd( accu, _smpy( coef[k][4], y));
    v2[k] = _sshl( accu, 1);

    x = y;
  }
  return( y);
}
```

D Algorithms

For typesetting algorithms/pseudocode, the `algorithm2e` package can be used, offering rich options. Algorithms can be referenced by usual cross-references, as here: see Alg. 1.

Algorithm 1: B-PHADQ

Choose parameters $\tau, \sigma, > 0,\ \rho \in [0, 1]$
Initialize $x^{(0)}, p^{(0)}, q^{(0)}$ and $\omega_{\mathbf{s}}$
for $i = 0, 1, \ldots$ **do**
  $q^{(i+1)} = \mathrm{clip}_{\lambda}(q^{(i)} + \sigma D R_{\omega_{\mathbf{s}}} G_g x)$
  $u = p^{(i)} - \tau G_g^* R_{\omega_{\mathbf{s}}}^* D^* q^{(i+1)}$ % auxiliary
  $p^{(i+1)} = \mathrm{proj}_{\Gamma}(u)$ % consistent
  $p^{(i+1)} = \frac{1}{\tau+1}(\tau\ \mathrm{proj}_{\Gamma}(u) + u)$ % inconsistent
  $x^{(i+1)} = p^{(i+1)} + \rho(p^{(i+1)} - p^{(i)})$
end
return $p^{(i+1)}$

E Content of the electronic attachment

An electronic attachment is often a part of the thesis. The attachment is uploaded in the BUT information system together with the thesis PDF. Please use an appropriate file format for the attachment.

It is suggested to comment on every folder, to specify which of the files contains main settings, to specify which is the main or executable file, what was the setting of the compiler etc. It is also valuable to specify in which version of the software the code has been tested (e.g. Matlab 2018b). In the case that hardware has been created within the thesis, the electronic attachment must contain all documentation (for example Eagle files with the printed circuit board layout).

If your attachment contains a lot of files or folders, LaTeX package `dirtree` can become handy, as in the following example.

```
/ ..... root of the attached archive
├── logo ..... logotypes
│   ├── BUT_abbreviation_color_PANTONE_EN.pdf
│   ├── BUT_color_PANTONE_EN.pdf
│   ├── FEEC_abbreviation_color_PANTONE_EN.pdf
│   └── UTKO_color_PANTONE_EN.pdf
├── pdf ..... PDFs (generate them in the information system)
│   ├── assignment-example.pdf
│   ├── cover-example.pdf
│   └── titlepage-example.pdf
├── pict ..... other graphic files
│   ├── soucastky.png
│   ├── spoje.png
│   ├── ZlepseneWilsonovoZrcadloNPN.png
│   └── ZlepseneWilsonovoZrcadloPNP.png
├── text ..... LaTeX source codes of the text
│   ├── abbreviation.tex
│   ├── appendix.tex
│   ├── bibliography.tex
│   ├── conclusion.tex
│   ├── introduction.tex
│   ├── results.tex
│   └── solution.tex
├── template-thesis.tex ..... main file of the thesis
├── template-presentation.tex ..... main file of the slides for presentation
└── thesis.sty ..... package for typesetting final theses at BUT
```