

# Improvements in space radiation-tolerant FPGA implementation of land surface temperature-split window algorithm

Assaad El Makhoulfi, Nisrine Chekroun, Noha Tagmouti, Samir El Adib, Naoufal Raissouni  
Remote Sensing and GIS Unit (RS&GIS), National School of Applied Sciences, University of Abdelmalek Essaadi,  
Tetuan, Morocco

## Article Info

### Article history:

Received Nov 16, 2020

Revised Mar 26, 2021

Accepted Apr 10, 2021

### Keywords:

FPGA

Land surface temperature

VHDL

Virtex

## ABSTRACT

The trend in satellite remote sensing assignments has continuously been concerning using hardware devices with more flexibility, smaller size, and higher computational power. Therefore, field programmable gate arrays (FPGA) technology is often used by the developers of the scientific community and equipment for carrying out different satellite remote sensing algorithms. This article explains hardware implementation of land surface temperature split window (LST-SW) algorithm based on the FPGA. To get a high-speed process and real-time application, VHSIC hardware description language (VHDL) was employed to design the LST-SW algorithm. The paper presents the benefits of the used Virtex-4QV of radiation tolerant series FPGA. The experimental results revealed that the suggested implementation of the algorithm using Virtex4QV achieved higher throughput of 435.392 Mbps, and faster processing time with value of 2.95 ms. Furthermore, a comparison between the proposed implementation and existing work demonstrated that the proposed implementation has better performance in terms of area utilization; 1.17% reduction in number of Slice used and 1.06% reduction in of LUTs. Moreover, the significant advantage of area utilization would be the none use of block RAMs comparing to existing work using three blocks RAMs. Finally, comparison results show improvements using the proposed implementation with rates of 2.28% higher frequency, 3.66 x higher throughput, and 1.19% faster processing time.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Assaad El Makhoulfi

Remote Sensing and GIS Unit (RS&GIS)

National School of Applied Sciences, University of Abdelmalek Essaadi

Tetuan, Morocco

Email: [assaadelmakhoulfi@gmail.com](mailto:assaadelmakhoulfi@gmail.com)

## 1. INTRODUCTION

Rapid prototyping of high-complexity digital circuits is now possible thanks to the density of current programmable circuits, such as field programmable gate arrays (FPGAs) [1]. It is possible to quickly test the validity of new architectural concepts: the complete implementation of a processor on FPGAs circuits is today within our range, resulting in more evaluation possibilities than those offered by software simulators. Moreover, the re-programmability of some FPGA chips has opened new research avenues: design methodologies for reconfigurable systems, able to evolve or adapt to varying environments or constraints. These developments are proposing novel opportunities, particularly in satellite remote sensing. Digital sensors mounted onboard remote sensing satellites examine massive regions of the Earth's surface day and

night and transmit these information to the ground aimed for processing and use in a variety of applications: control systems engineering, bioinformatics, land use [2], land cover [3] aerospace and defense systems.

Remote sensing satellites are increasingly being used and use the latest available technical components as a result of their continuous development and improvement [4]. The appearance of FPGAs [5], can facilitate on-board and real-time remote sensing data analysis. Fortunately, related studies have released various algorithms in the area of remote sensing implementing using FPGAs such as; i) real-time marker-based visual sensor based on an FPGA and soft-core processor [6], ii) FPGA implementation of an algorithm for automatically detecting targets [7], iii) FPGA implementation of the N-FINDR algorithm [8], iv) novel FPGA-based architecture for fast automatic target detection in hyperspectral images [9]. These research works are altogether based on implementations using different types of FPGAs, nevertheless these last are not specific neither for remote sensing satellite missions nor are designed for space radiation environment. Moreover, satellite remote sensing new missions require potent FPGAs with significant resources and consuming low amounts of energy. Besides, FPGAs are absolutely required to reach higher throughput with maximum frequency. FPGAs provide many advantages to space applications [10] reaching excellent productivity and integration without incurring costs, risk, and long development times of application-specific integrated circuits (ASICs), introducing hardware improvements late in the design cycle and ensuring stable operation in high radiation environments.

Small satellite missions that characterized by capable low-cost platforms with budget implemented and flexibility schedules, have been the focus of spatial innovation in recent years. In reality, Earth observation satellites face significant challenges because the land surface is literally heterogeneous, consisting of a variety of materials with varying geometrical properties [11]. Land surface temperature (LST) is valuable parameter associated with surface energy and water balance in local and international scales. Actually, one of the most significant biophysical parameter measured by means of remote sensing satellites is surface temperature [12].

Hardware devices with small size and cost, as well as flexibility and high computing capacity, have been widely used in the design of hardware modules for remote sensing missions. For instance, as a solution, on-board processing allows a great reuse of expensive hardware resources. Consequently, FPGAs have many advantages that make them attractive for various applications. FPGAs are fully reprogrammable [13], meaning that even after the circuit has been designed and implemented, they can be modified, updated, and their functionality can be completely altered to perform different tasks. Furthermore, an FPGA performs better than a formal central processing unit (CPU) since this reprogrammable circuit integrates a significant number of arithmetic blocks that can be low-complexity blocks (e.g., simple multipliers), or can be relatively complex (e.g., digital signal processor (DSP) units) consisting of combinations of different components (e.g., multipliers, adders, accumulators and shift registers). These DSPs have the advantage of accelerating the FPGA's computing performance and enables them to achieve greater efficiency and flexibility, while reducing costs and power consumption [14]. Moreover, the reconfigurable hardware was accredited by international remote sensing organizations and was widely employed in missions of remote sensing [15] and space-borne Earth observations [16]. Integrating high processing power into an embedded system typically involves heterogeneous, parallel architectures to find the best tradeoffs in performance, size, and consumption. Since the FPGAs are capable of implementing a variety of algorithms, they are useful for several missions and enable the algorithm for space computers to be modified after implementation to satisfy novel mission needs or to eliminate design flaws.

Being radiation hardened, Xilinx's static random-access memory (SRAM)-based FPGAs and microsemi's flash-based FPGAs are those mainly commonly employed in space. The optimization of the algorithms to the architecture of the component (i.e., parallelism and look up tables) in some cases increases the performance to exceed those of implementation on DSP and ASICs. Besides, the description of an architecture optimized in very-high-speed integrated circuit hardware description language (VHDL) synthesizable code with a hierarchical and modular approach provides good flexibility to the system [17].

The proposed implementation presents a noteworthy improvements on the results shown in the research paper of Raissouni *et al.* [18] by means of integrating a radiation-tolerant FPGAs. Besides, when compared with recent state-of-the-art results, our optimal method provides an improvement in throughput while consuming less FPGA resources. Further, the advantage of using radiation-tolerant FPGA in the proposed implementation for satellite remote sensing missions, is leading to promising performance that is suitable for future satellite remote sensing missions and CubeSats on board land surface temperature (LST) computations purposes.

The foremost objective of the current research is to develop an FPGA-based hardware version of Sobrino and Raissouni [12] land surface temperature split-window (LST-SW) algorithm, operating in high radiation environments. The space-grade Virtex-4QV FPGA was designated for this purpose, since it is suitable for a space environment with radiation. Thus, through this implementation, a high throughput was

achieved with higher frequency and a less resource was consumed. Moreover, the proposed implementation has the benefit to be used for satellites systems to be designed using similar thermal infrared channels.

Section 2, of this research paper presents the purpose of using radiation-tolerant FPGA in satellite remote sensing applications. Section 3, describes an overview of the LST-SW algorithm. Section 4, is illustrating the proposed hardware FPGA architecture for the LST-SW algorithm. Finally, section 5 analyses the implementation results including the resources used, throughput and timing.

## 2. PURPOSE OF USING RADIATION-TOLERANT FPGA IN SATELLITE REMOTE SENSING

The use of reconfigurable hardware in space-based applications for remote sensing is increasingly of interest [19], [20]. Integrating FPGAs in a spacecraft enables application-specific hardware to be used among smaller size, lower cost, increased flexibility and higher computational capacity. As the downlink bandwidth does not keep pace, higher rises in sensor resolution in remote sensing space payloads produce a bottleneck in processing [21]. Operators need on-board processing to send the processed information to the satellites, not just raw data. This is a producing challenge for the approximately 100 remote sensing satellites launched each year. Nowadays, FPGA technology is being projected as a vital alternative: Prior to downlink, remote sensing data can be processed and interpreted in orbit-instead of storing and transmitting entirely the captured data/images-resulting in a significant bandwidth reduction. Subsequently, calculations to be carried out at ground stations come to be easier and faster [22].

Electronic devices external the Earth's atmosphere are usually subjected to a different radiation environment than the one on Earth. The activity of a traditional semiconductor system may be affected or interrupted by high levels of radiation. Electronic circuits can be built with specialized manufacturing techniques tolerating high radiation rates. With an increased focus on the exploitation of programmable logic in spatial uses, numerous researchers have examined the appropriateness of commercially available FPGAs in radiation environments [23]. The FPGAs based on the Xilinx SRAM and Microsemi's flash FPGAs are currently being used to address the problem by incorporating high-speed signal processing and built-in radiation reduction methods in order to keep the devices in difficult radiation environments operational. Furthermore, the FPGAs maintain low static power and reduce dynamic energy requirements significantly. Over 150,000 logical elements and a device efficiency of up to 300 MHz are available in this latest class of FPGAs for radiation-tolerant in a substantially higher proportion than other radiation-resistant FPGA technologies, including combinatory logic, DSP math blocks and transceivers.

SRAM-based FPGAs offer high performance, high logic density, and low non-recurring engineering (NRE) costs when compared to other FPGA technologies. At the same time, FPGAs can be statically reconfigured an almost infinite number of times after the initial power-on setup. In many applications, the advantages of SRAM-based FPGAs are considered to be dominant. In terms of flash technology, its total ionizing dose (TID) limitations and potential charge leakage is undergoing scrutiny by the industry. Furthermore, it does not currently support dynamic partial reconfiguration. Finally, the Xilinx Virtex-QV is found to have the greatest logic density, performance, and radiation-tolerance of all SRAM options, combining high-speed signal processing with special built-in radiation mitigation techniques to keep systems operational in harsh radiation environments. Radiation-hardened FPGAs are, in fact, in high demand for military and space applications [24]. Another reason is that Xilinx has introduced many radiation-tolerant FPGAs, including the space-grade Virtex-QV line of high-reliability FPGA chips with million gate densities to support remote sensing applications' high throughput requirements [25].

The space-grade Virtex-QV uses hardware and package hardening techniques, these particular reliability-enhancing techniques are: i) SRAM transistor cells for memory and latches configuration, ii) Triple modular redundancy (TMR) for configuration control, iii) Single event transient (SET) filter option for the configuration logic blocks (CLBs), and iv) epitaxial layer and protection layers. On the other hand, radiation hardness of the Virtex-QV comes at a higher energy consumption price [26].

In this work, we make use of the Xilinx space-grade Virtex-4QV XQR4VSX55 FPGA [27] for the proposed implementation of the LST-SW algorithm. The Virtex-4QV is the industry's high-performance radiation-hardened reconfigurable FPGA for processing-intensive space systems. The device offers one of the highest densities, performance and integration capabilities enabling more complex and capable systems over radiation-hardened ASIC devices with their high development costs and long lead-time. The Virtex-4QV is thoroughly tested for radiation-tolerance and is demonstrated to tolerate a total dose in the range of 300 krads, which is more than acceptable for many space applications.

### 3. LST-SW ALGORITHM

Land surface temperature (LST) occupy an important role in the land surface features at the local and global scale and is one of the most key variable in the biophysics of land surface procedures [28]. LST is determined from the use of either empirical or physical algorithms of satellite measurements in the thermal infrared (TIR) spectral bands. Numerous LST algorithms have been expounded and described over time in literature [29]-[34]. An amount of Earth observation satellites (e.g., NOAA/AVHRR, TERRA/MODIS, LANDSAT/TM and ENVISAT/AATSR) have TIR channels on thermal sensors to derive LST.

The estimation of LST from satellite information is mostly affected by the atmosphere and surface emissivity [35]. We apply the operational LST-SW algorithm proposed by Sobrino and Raissouni [12] to make accurate estimations of this parameter:

$$T_s = T_4 + 1.40(T_4 - T_5) + 0.32(T_4 - T_5)^2 + 0.83 + (57 - 5W)(1 - \epsilon) - (161 - 30W)\Delta\epsilon \quad (1)$$

where  $T_s$  is the LST,  $T_4$  is the brightness temperatures measured in AVHRR channels 4, and  $T_5$  is the brightness temperatures measured in AVHRR channels 5,  $\epsilon = 0.5(\epsilon_4 + \epsilon_5)$  and  $\Delta\epsilon = (\epsilon_4 - \epsilon_5)$  are, respectively, the average effective emissivity in channels (4,5) and the spectral variation of emissivity [36]. To calculate the total amount of atmospheric water vapor,  $W$  (g cm<sup>-2</sup>), we used the approach named the split-window covariance-variance ratio (SWCVR) [37]. This approach allows the estimation of  $W$  from only satellite data [38].

### 4. THE PROPOSED HARDWARE FPGA ARCHITECTURE FOR THE LST-SW ALGORITHM

The parallel processing of an FPGA is key research guidance of the fast calculation community with high efficiency. Different factors such as the logical resource level in the chosen FPGA, and the optimized nature of algorithms affect its calculation speed [39]. Figure 1 displays the corresponding block diagram of the proposed Xilinx Virtex-4QV FPGA LST-SW implementation: i) the input data are set as  $T_4$ ,  $T_5$ ,  $W$  and Epsilon corresponding to the  $T_4$ ,  $T_5$ ,  $W$  and  $\epsilon$  satellite data, and ii) the output result of the LST-SW algorithm is set as LST. Thus, considering both the FPGA hardware architecture (i.e., four FIFOs instead of five FIFOs), and the study area characteristics (see § 5.1), we consider in our case  $\Delta\epsilon = 0.005$ .

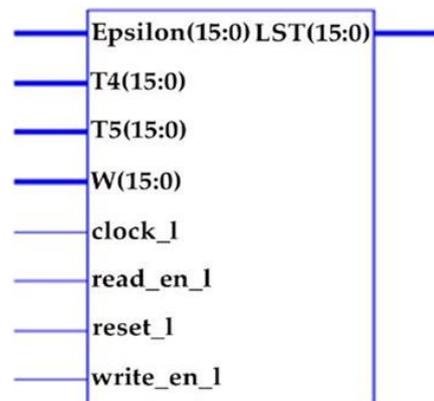


Figure 1. Xilinx virtex-4QV FPGA LST-SW block diagram controlled by clock\_1, reset\_1, read\_en\_1, and write\_en\_1

Accordingly, the proposed architecture is involving two parts:

- LST1 (Part 1), as shown in (2), computing the main part of the LST algorithm (1) and
- LST2 (Part 2), as shown in (3), computing the correction part of the LST algorithm (1).

LST is then computed as the sum of LST1 and LST2, as shown in (4).

$$LST1 \text{ (Part 1)} = T_4 + 1.40(T_4 - T_5) + 0.32(T_4 - T_5)^2 + 0.83 \quad (2)$$

$$LST2 \text{ (Part 2)} = (57 - 5W)(1 - \epsilon) - (161 - 30W)\Delta\epsilon \quad (3)$$

$$LST = LST1 + LST2 \quad (4)$$

Figure 2 shows the corresponding general hardware architecture. Satellite data/raw images [i.e.,  $T_4$  (in K,  $\times 10$ ),  $T_5$  (in K,  $\times 10$ ),  $W$  (in g cm $^{-2}$ ,  $\times 1000$ ), and  $\epsilon$  ( $\times 1000$ ), see § 5.1 for more details on data format] are saved on the hard disk and transferred to each considered part [i.e., LST1 (Part 1), and LST2 (Part 2)] via the corresponding FIFOs and variables. FIFO1 and FIFO2 transmit  $T_4$ ,  $T_5$  respectively to LST1 (Part 1), FIFO3 and FIFO4 transmit  $W$  and  $\epsilon$  respectively to LST2 (Part 2). Finally, LST1 and LST2 are re-transmitted via FIFO1 and FIFO2 respectively to be added. Figure 3 provides the equivalent structural description of the proposed FPGA LST-SW hardware in VHDL language. In fact, we only included the instantiations required for the main components and left out some components, signal definitions, and interconnection details.

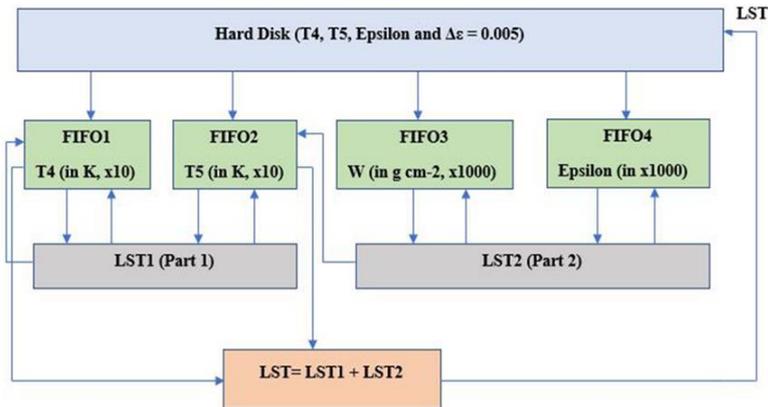


Figure 2. Hardware architecture proposed for the implementation of the FPGA LST-SW algorithm

```
entity LST_module is
port(
clock_I : in std_logic; reset_I : in std_logic; read_en_I:in std_logic; write_en_I:in std_logic;
T4: in std_logic_vector(15 downto 0);
T5: in std_logic_vector(15 downto 0);
W: in std_logic_vector(15 downto 0);
Epsilon: in std_logic_vector(15 downto 0);
LST :out std_logic_vector(15 downto 0));
end LST_module;
architecture Behavioral of LST_module is

COMPONENT LST1(Part1)
COMPONENT LST2(Part2)

--others components--
--Signals definition--

begin
LST_module1: LST1(Part1)
port map(clocka=>clock_I, reseta=>reset_I, read_ena=>read_en_I,
write_ena=>write_en_I, outlar=>T4, outlar2=>T5, LST1=>int_signal);

LST_module2: LST2(Part2)
port map(clocka=>clock_I, reseta=>reset_I, read_ena=>read_en_I,
write_ena=>write_en_I, outlar3=>W,outlar4=>Epsilon, LST2=>int_signal2);

LST_module3: LST_FIFO
port map(clock=>clock_I, reset=>reset_I, read_en=>read_en_I, write_en=>write_en_I,
FIFO_1=>int_signal, A=>int_signal3, FIFO_2=>int_signal2, B=>int_signal4);
```

Figure 3. VHDL description language for the LST-SW hardware module, this structural description contains three main component (LST1 (Part1), LST2 (Part2) and LST\_FIFOs)

Consequently, for computational reasons (i.e., to preserve both the integer format and size and the decimal precision for temperature data in Kelvin,  $T_4$  and  $T_5$ ), we proceed by: (i) dividing LST1 coefficients by 10 as shown in (5) and; (ii) dividing LST2 coefficients by 1000 as shown in (6).

$$\text{LST1 (Part 1)} = 0.1 T_4 + 0.14 (T_4 - T_5) + 0.0032 (T_4 - T_5)^2 + 0.83 \quad (5)$$

$$LST2 \text{ (Part 2)} = [(57000 - 5 W) (1000 - \epsilon) - (161000 - 30 W) \times 5] \times 0.001 \tag{6}$$

We can see from Figure 4 the architecture used in the proposed design to implement LST1 and LST2 modules. In the entire procedure, image data are saved in memory separately and released into the system pixel by pixel. The processing to compute the first pixel result of the LST is: FIFO1 and FIFO2 read the first pixel of  $T_4$  and  $T_5$  respectively, then transmitted to the first part to be calculated as show in (a) and is similar to the first part of the (2), and with the same time FIFO3 and FIFO4 read the first pixel of  $W$  and Epsilon respectively, then transmitted data to the second part to be calculated as show in (b) and is similar to the first part of the (3). Finally, as in Figure 2 we add the two part: LST1 and LST2. We replicate this operation in the whole procedure line by line until the pixels finish.

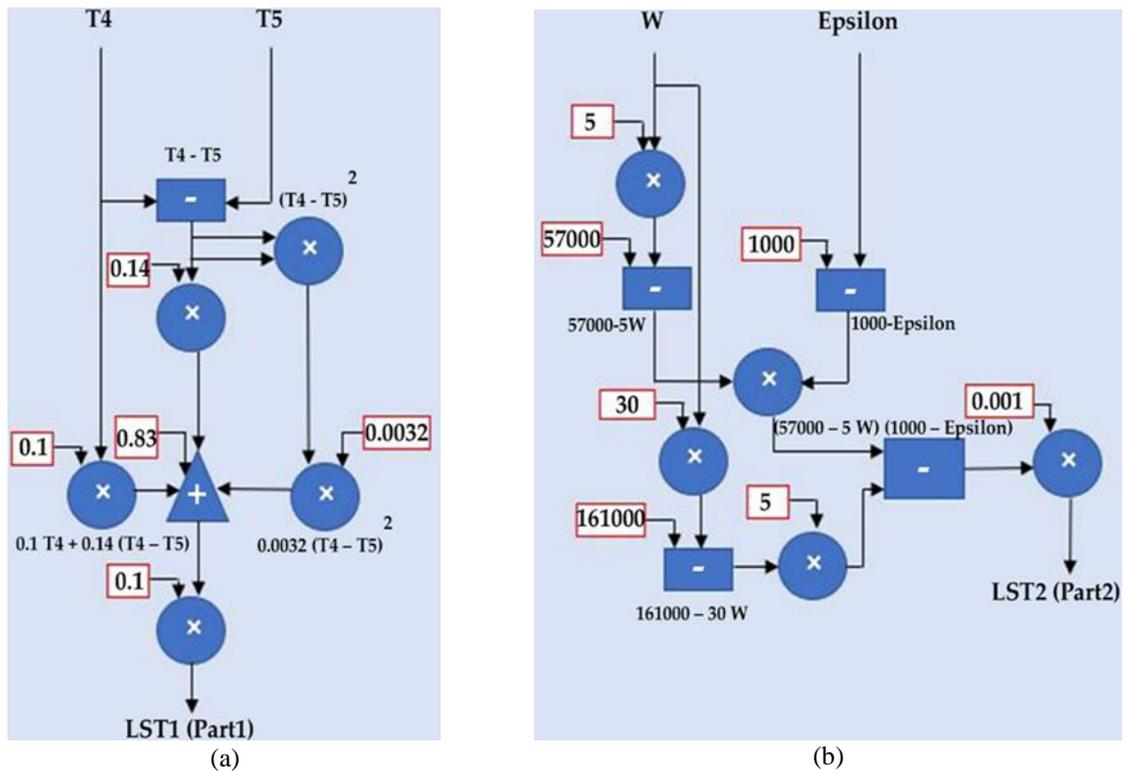


Figure 4. Hardware architecture used to implement the two module LST1 and LST2, (a) Hardware architecture adopted to implement LST1 module, (b) Hardware architecture adopted to implement LST2 module

The fixed-point package was choosing in the arithmetic logic units to reduce area utilization and get a better precision. In fact, even though the type REAL specified in the package standard has limited synthesis support, newer options for dealing with fixed point was used in the proposed implementation and represented by using the library fixed\_pkg in the VHDL code for the two modules LST1 and LST2. In arithmetic logic units we chose to use a fixed-point package for decimal numbers in the proposed algorithm ((18) and (19)) instead of floating point. In fact, fixed point arithmetic is valuable as it results in faster and smaller functional units. However, if not carefully constructed, it can produce less accurate results. On the other hand, floating point arithmetic is consuming in terms for hardware and leads to inefficient designs, especially for FPGA implementation.

The addition, subtraction, multiplication, and division operations are required by the FPGA LST-SW implementation algorithm as shown in Figure 4. The number of calculations increases as the resources increase. Several of these computations are straightforward, such as addition, subtraction and multiplication; however, a division operation is complex and can affect the precision significantly. Therefore, in the proposed implementation we handled this carefully to prevent overflow, which leads to incorrect results using the functions existed in the library fixed\_pkg. For the division operation, instead of dividing by 10, we have multiplied the result of the LST1(Part1) by 0.1. Similarly, we have multiplied the result by 0.001 in the

LST2(Part2). This has a positive effect, as the multiplication does not consume many resources and does not affect precision.

Figure 5 shows the register transfer level of the FPGA LST-SW implementation algorithm and the connection between different blocks, with 16 bits range of data that sent it into the system as the output LST result. Furthermore, Figure 5 shows the input signals T4, T5, W and Epsilon. Additionally, the output of each part of LST1 and LST2 are connecting to FIFO1 and FIFO2 respectively. In addition, each component (LST1(part1), LST2(Part2) and FIFOs) of the proposed implementation. is controlled by clock, reset, read, and write.

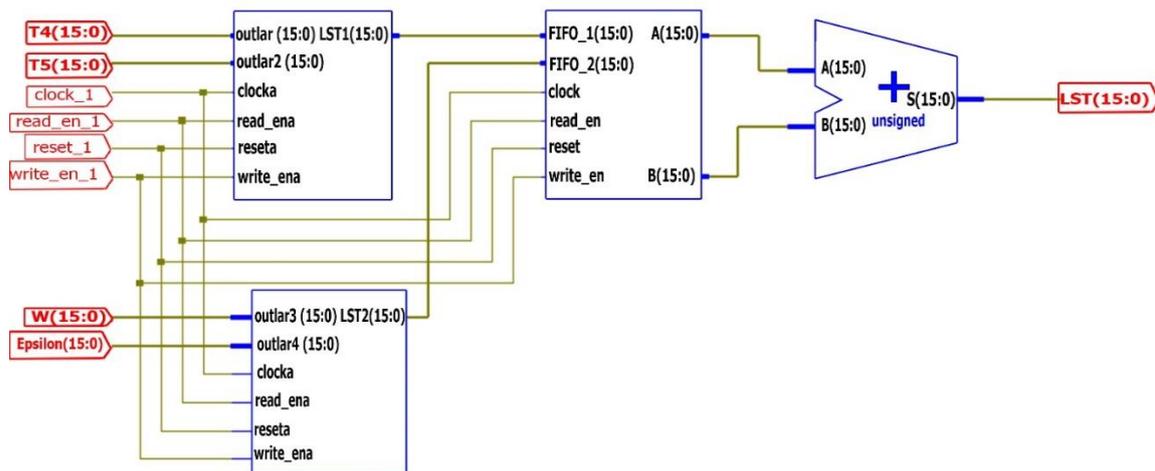


Figure 5. Internal register transfer logic for the system proposed

## 5. RESULTS

### 5.1. Study area

The pathfinder AVHRR land (PAL) satellite dataset has been exploited for examine the Mediterranean basin. The calibration of the imageries is attained by the recommendations advocated by the NOAA [40], [41]. Table 1 illustrates the characteristics of the cloud-free images used in this application and corresponds to the period of July 1982.

Table 1. Characteristics of the cloud-free images used in this application, corresponding to the input data: T4, T5, W and epsilon of the FPGA LST-SW implementation

AVHRR Data	Description	Format	Pixels	Type	Projection	Unit
T <sub>4</sub>	Radiance temperature for channel 4	695×316	219,620	Integer	Homolosine	K (x10)
T <sub>5</sub>	Radiance temperature for channel 5	695×316	219,620	Integer	Homolosine	K (x10)
W	Total amount of atmospheric water vapor	695×316	219,620	Integer	Homolosine	g cm <sup>-2</sup> (x1000)
ε	Effective emissivity in both channels 4 and 5	695×316	219,620	Integer	Homolosine	x1000

### 5.2. FPGA LST-SW simulation results

In this part, we present an experimental result of the proposed FPGA LST-SW implementation algorithm's computational efficiency. For the specification of the LST module, the hardware architecture defined in Section 3 was implemented using the VHDL language. Furthermore, we specified the entire system using the Xilinx ISE 14.7 environment. Table 2 shows the resources consumed by the proposed LST-SW algorithm design, using the space-grade Virtex-4QV XQR4V5X55. This FPGA has a total of 24,576 slices, 49,152 slice flip-flops, and 49,152 four-input LUT available. Moreover, the FPGA includes some heterogeneous resources, such as DSP48s, and has a total of 512 of them. Moreover, the reason for working with this type of device (XQR4V5X55) in the algorithm LST is because the latter contains too many arithmetic operations. For this reason, we need several DSP plus this type, especially for the optimization of the ultra-efficient signal processing. We used these resources to optimize the design in the proposed implementation.

Table 2. Summary of resource utilization for the proposed FPGA-based implementation of the LST algorithm

Target FPGA Device utilization	Virtex-4QV SX55		
	Used	Total	%
Number of Slice	1124	24,576	4
Number of Slice Register	2120	49,152	4
Number of Slice LUTs	2438	49,152	4
Number of DSPs	16	512	5
Number of bonded IOBs	84	640	13
Maximum Frequency	190.484 MHz		

The block slice registers were employed to implement the FIFOs without using block RAMs. With other slices are used for the implementation of the LST-SW algorithm. Moreover, the number of DSP is higher because of the existence of the calculation in each part (LST1 and LST2), which consumes much of the resources of the FPGA. The time used in the implementation allowed us to achieve a maximum frequency of 190.484 MHz. Figure 6 displays the LST-SW images computed in the period July using the proposed implementation. In complete coherence with the biodiversity of the same, a large LST variability is seen for the area and period.

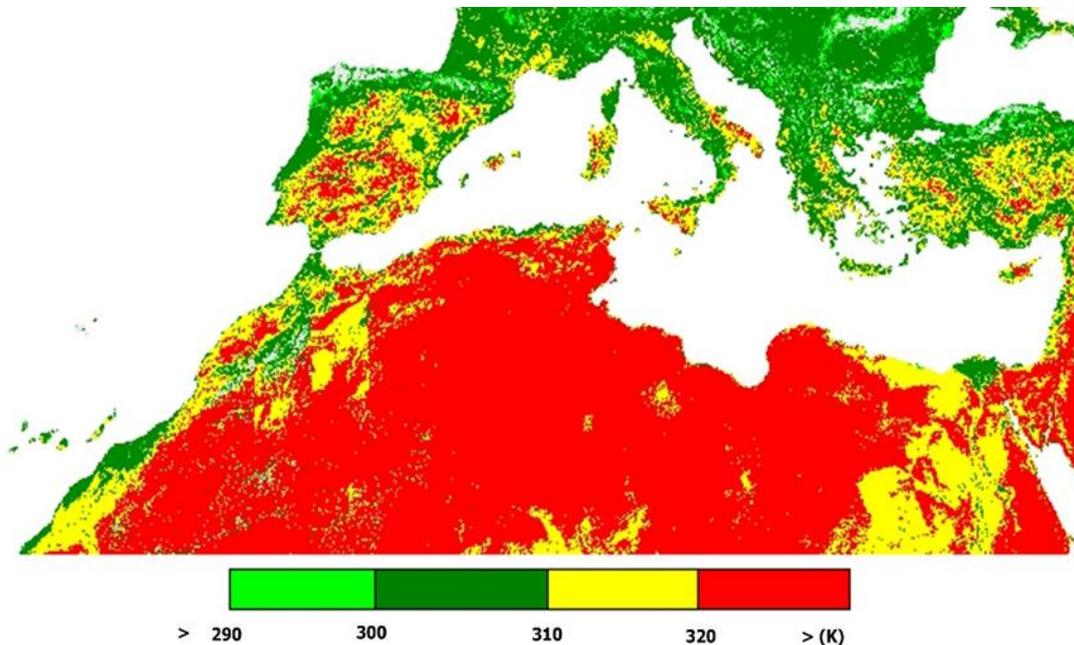


Figure 6. LST-SW images computed using the proposed FPGA based hardware version during July 1982

### 5.3. Comparison of performance between the proposed LST-SW existing implementation

We evaluate the performance of the proposed implementation by performing a comparison with LST-SW implementation that has already been implemented in [18] as describing in Table 3. The metric for evaluating the performance of the proposed LST implementation comparing to the existing work was based on the resources used, the frequency, the throughput, Throughput per slice (TPS), and the processing time. Other features such as power consumption are not describing in this implementation. Table 3 presents the comparison of performance between the proposed architecture and the existing architecture [18].

Table 3. Performance comparison between the proposed implementation and existing design

Design	Device	Number of slices used	Number of Slice LUTs used	Block RAMs	Number of DSPs used	Fmax (MHz)	Throughput (MBps)	Throughput per Slice
Ref. [18]	Virtex-5 LX50	1169	2590	3	16	83.330 MHz	118.700	0.101
Proposed	Virtex-4 QVXS55	1124	2438	-	16	190.484 MHz	435.392	0.387

As we can see, the proposed implementation in terms of hardware consuming is much better than the existing implementation. In fact, the proposed design uses 1124 slices from a total of 24,576 slices and 2438 slice LUTs from a total of 49,152 compared with 1169 slices from a total of 7200 slices and 2590 slice LUTs from a total of 28,800 in [18]. Moreover, the way in which FIFOs were implemented in the proposed implementation and in [18] is different. The FIFOs used in the proposed architecture were implemented with the slices, but in [18] they implemented FIFOs based on the Blocks RAMs, so there is a benefit of the proposed architecture here in terms of resource optimization. In addition, we notice from Table 3 that the frequency of the proposed architecture is higher than the frequency obtained in [18], which leads to faster response times and high-speed calculation of the LST algorithm. Another important metric is the throughput, and signifies the number of bits processed per unit time and is specified in Gbps or Mbps. The throughput is calculated using (7).

$$\text{Throughput} = \frac{\text{Number of bits processed} \times F_{\max}}{\text{Latency}} \quad (7)$$

In the proposed architecture, the number of bits processed is 16,  $F_{\max}$  is the tool's maximum frequency, and Latency is the number of clock cycles after which output is generated, which is equal to seven clock cycles. Therefore, the proposed design achieves the highest throughput of 435.392 Mbps on Virtex-4QV with a maximum clock frequency of 190.484 MHz compared with reference [18]. In order to measure the hardware resource cost associated with implementations resultant throughput, TPS metric was applied by using (8):

$$\text{TPS} = \frac{\text{Throughput}}{\text{CLB Slices Used}} \quad (8)$$

The CLB is an acronym for Configurable Logic Block. As shown in Table 3, the number of slices used is 1124, so the TPS of the proposed architecture is about 0.387 Mbps/Slices, which is higher than the current implementation in [18]. Finally, Table 4 shows the processing times obtained using the considered FPGA implementation [18], as well as an analogous software version developing in LabVIEW and run on an embedded real-time controller with a 533 MHz PowerPC, 256 MB of DDR2 RAM, and 2 GB of non-volatile storage [18].

Table 4. Processing times estimated for the proposed hardware implementation and for an equivalent software version and for the current implementation in [18]

	Processing Time (ms)
Ref. [18]	3.53
Proposed (Virtex-4QV SX55)	2.95
Ref. [18]	11.14

As can be observed from Table 4, the processing time in the proposed architecture is faster compared to the software version and from [18]. Another important aspect in the hardware implementation is the issue of arithmetical precision. As we said in Section 3, we have paid special attention to this problem in our design. The fixed-point structure helps us reduce processing time and use less logical resources in the proposed implementation of the LST-SW algorithm using the fixed\_pkg library in VHDL. As a result, the proposed implementation is a major step forward in ensuring that the LST-SW algorithm is properly used in scenarios requiring real-time processing.

## 6. CONCLUSION

From the perspective of remote sensing systems, the reconfigurability of FPGA systems opens up a lot of innovative possibilities. This ranges from the appealing possibility of choosing the data processing algorithm to be used aboard, out of a set of algorithms that are available, from the Earth control station immediately following the data collection from the sensor to the potential one. Radiation-hardened FPGAs can be easily mounted or embedded in the sensor due to their compact size and low weight, will greatly benefit current sensor design practices. The role of FPGAs in remote sensing missions was discussed in this paper, as well as the benefits of using FPGAs, such as being radiation-hardened, the experimental results of the FPGA implementation of the LST-SW algorithm, and the performance of the proposed architecture in comparison to previous work. The results showed that the suggested implementation of the LST-SW using

Xilinx Virtex4QV XQR4VSX55 FPGA attained a higher throughput (435.392 Mbps) with frequency of 190.484 MHz and faster processing times, with values of 2.95 ms. In addition, the proposed implementation shows better performance in terms of area utilization compared to the existing work: 1.17% reduction in number of Slice used, a 1.06% reduction in number of Slice LUTs, and total elimination of block RAMs. Furthermore, the proposed implementation has 2.28% higher frequency and achieved higher throughput  $3.66 \times$  compared to the existing work, and faster processing times with decrease of 1.19 %, which can allow us to implement other algorithms with the same device. Furthermore, this implementation offers significant and promising performance, making it suitable for future CubeSats on board LST-SW computation. In addition, since the FPGA we used is especially radiation-hardened, the proposed implementation can be used, proposed, and even programmable for all CubeSats.

## ACKNOWLEDGEMENTS

The authors wish to express their gratitude for the data that used in this paper to the advanced very-high resolution radiometer (AVHRR) instrument, and the National Oceanic and Atmospheric Administration (NOAA) satellite platforms.

## REFERENCES

- [1] W. Li *et al.*, "A Real-Time Tree Crown Detection Approach for Large-Scale Remote Sensing Images on FPGAs," *Remote Sensing*, vol. 11, no. 9, pp. 1–20, Jan. 2019, doi: 10.3390/rs11091025.
- [2] H. Elmannai and A. D. Algarni, "Classification using semantic feature and machine learning: land-use case application," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 19, no. 4, Aug. 2021, doi: 10.12928/telkomnika.v19i4.18359.
- [3] H. Hashim, Z. A. Latif, and N. A. Adnan, "Land use land cover analysis with pixel-based classification approach," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 16, no. 3, pp. 1327–1333, Dec. 2019, doi: 10.11591/ijeecs.v16.i3.pp1327-1333.
- [4] G. Kaplan and U. Avdan, "Monthly Analysis of Wetlands Dynamics Using Remote Sensing Data," *ISPRS International Journal of Geo-Information*, vol. 7, no. 10, pp. 1–20, Oct. 2018, doi: 10.3390/ijgi7100411.
- [5] C. Yang *et al.*, "A Spaceborne Synthetic Aperture Radar Partial Fixed-Point Imaging System Using a Field-Programmable Gate Array—Application-Specific Integrated Circuit Hybrid Heterogeneous Parallel Acceleration Technique," *Sensors (Basel)*, vol. 17, no. 7, Jun. 2017, doi: 10.3390/s17071493.
- [6] H. Tayara, W. Ham, and K. T. Chong, "A Real-Time Marker-Based Visual Sensor Based on a FPGA and a Soft Core Processor," *Sensors*, vol. 16, no. 12, Dec. 2016, doi: 10.3390/s16122139.
- [7] C. González, S. Bernabé, D. Mozos, and A. Plaza, "FPGA Implementation of an Algorithm for Automatically Detecting Targets in Remotely Sensed Hyperspectral Images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 9, pp. 4334–4343, Sep. 2016, doi: 10.1109/JSTARS.2015.2504427.
- [8] C. Gonzalez, D. Mozos, J. Resano, and A. Plaza, "FPGA Implementation of the N-FINDR Algorithm for Remotely Sensed Hyperspectral Image Analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 2, pp. 374–388, Feb. 2012, doi: 10.1109/TGRS.2011.2171693.
- [9] J. Lei *et al.*, "A Novel FPGA-Based Architecture for Fast Automatic Target Detection in Hyperspectral Images," *Remote Sensing*, vol. 11, no. 2, Jan. 2019, doi: 10.3390/rs11020146.
- [10] M. R. Maheshwarappa, M. D. J. Bowyer, and C. P. Bridges, "Improvements in CPU FPGA Performance for Small Satellite SDR Applications," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 1, pp. 310–322, Feb. 2017, doi: 10.1109/TAES.2017.2650320.
- [11] N. Tajudin, N. Ya'acob, D. M. Ali, and N. A. Adnan, "Soil moisture index estimation from landsat 8 images for prediction and monitoring landslide occurrences in Ulu Kelang, Selangor, Malaysia," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2101–2108, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2101-2108.
- [12] J. A. Sobrino and N. Raissouni, "Toward remote sensing methods for land cover dynamic monitoring: Application to Morocco," *International Journal of Remote Sensing*, vol. 21, no. 2, pp. 353–366, Jan. 2000, doi: 10.1080/014311600210876.
- [13] G. Singh, "Reconfigurable Computing A review of the technology and its architecture," *IOSR Journal of VLSI and Signal Processing*, vol. 3, no. 4, pp. 08–14, 2013, doi: 10.9790/4200-0340814.
- [14] D. Wang, K. Xu, J. Guo, and S. Ghiasi, "DSP-Efficient Hardware Acceleration of Convolutional Neural Network Inference on FPGAs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4867–4880, Dec. 2020, doi: 10.1109/TCAD.2020.2968023.
- [15] G. Lagerloef *et al.*, "The Aquarius/SAC-D Mission: Designed to Meet the Salinity Remote-Sensing Challenge," *Oceanog.*, vol. 21, no. 1, pp. 68–81, Mar. 2008, doi: 10.5670/oceanog.2008.68.
- [16] M. M. Verstraete, D. J. Diner, and J.-L. Bézy, "Planning for a spaceborne Earth Observation mission: From user expectations to measurement requirements," *Environmental Science & Policy*, vol. 54, pp. 419–427, Dec. 2015, doi: 10.1016/j.envsci.2015.08.005.
- [17] V. Guddad, A. Kourfali, and D. Stroobandt, "VHDL design tool flow for portable FPGA implementation," *The Twelfth International Conference on Advances in Circuits, Electronics and Micro-electronics*, 2019, pp. 1–6.

- [18] N. Raissouni *et al.*, "Towards LST split-window algorithm FPGA implementation for CubeSats on-board computations purposes," *International Journal of Remote Sensing*, vol. 40, no. 5–6, pp. 2435–2450, Mar. 2019, doi: 10.1080/01431161.2018.1562589.
- [19] A. EL Makhloufi, S. EL Adib, and N. Raissouni, "An approach for FPGA implementation of LST-SW algorithm," *presented at the Third International Conference on Computing and Wireless Communication Systems, ICCWCS 2019*, Faculty of Sciences, Ibn Tofail University -Kénitra- Morocco, May 2019. [Online]. Available: <https://eudl.eu/doi/10.4108/eai.24-4-2019.2284220>.
- [20] G. M. Almeida, E. A. Bezerra, L. V. Cargnini, R. D. R. Fagundes, and D. G. Mesquita, "A Reed-Solomon Algorithm for FPGA Area Optimization in Space Applications," in *Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007)*, Aug. 2007, pp. 243–249, doi: 10.1109/AHS.2007.17.
- [21] M. Surratt, H. H. Loomis, A. A. Ross, and R. Duren, "Challenges of Remote FPGA Configuration for Space Applications," in *2005 IEEE Aerospace Conference*, Mar. 2005, pp. 1–9, doi: 10.1109/AERO.2005.1559549.
- [22] H. J. Kramer and A. P. Cracknell, "An overview of small satellites in remote sensing," *International Journal of Remote Sensing*, vol. 29, no. 15, pp. 4285–4337, Aug. 2008, doi: 10.1080/01431160801914952.
- [23] L. Rockett, D. Patel, S. Danziger, B. Cronquist, and J. J. Wang, "Radiation Hardened FPGA Technology for Space Applications," in *2007 IEEE Aerospace Conference*, Mar. 2007, pp. 1–7, doi: 10.1109/AERO.2007.353098.
- [24] P. Lysaght, B. Blodget, J. Mason, J. Young, and B. Bridgford, "Invited Paper: Enhanced Architectures, Design Methodologies and CAD Tools for Dynamic Reconfiguration of Xilinx FPGAs," in *2006 International Conference on Field Programmable Logic and Applications*, Aug. 2006, pp. 1–6, doi: 10.1109/FPL.2006.311188.
- [25] M. J. Wirthlin, "FPGAs operating in a radiation environment: lessons learned from FPGAs in space," *J. Inst.*, vol. 8, no. 02, pp. C02020–C02020, Feb. 2013, doi: 10.1088/1748-0221/8/02/C02020.
- [26] R. Glein *et al.*, "Reliability of space-grade vs. COTS SRAM-based FPGA in N-modular redundancy," in *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, Jun. 2015, pp. 1–8, doi: 10.1109/AHS.2015.7231159.
- [27] XILINX, "Space-Grade Virtex-4QV Family Overview (DS653)," 2014. [Online]. Available: <https://www.xilinx.com/products/silicon-devices/fpga/virtex-4qv.html>.
- [28] N. Morrow and M. A. Friedl, "Modeling biophysical controls on land surface temperature and reflectance in grasslands," *Agricultural and Forest Meteorology*, vol. 92, no. 3, pp. 147–161, Oct. 1998, doi: 10.1016/S0168-1923(98)00098-7.
- [29] J. A. Sobrino, N. Raissouni, and Z. L. Li, "A Comparative Study of Land Surface Emissivity Retrieval from NOAA Data," *Remote Sensing of Environment*, vol. 75, no. 2, pp. 256–266, Feb. 2001, doi: 10.1016/S0034-4257(00)00171-1.
- [30] Z. Wan and J. Dozier, "A generalized split-window algorithm for retrieving land-surface temperature from space," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 4, pp. 892–905, Jul. 1996, doi: 10.1109/36.508406.
- [31] Z. Qin, A. Karnieli, and P. Berliner, "A mono-window algorithm for retrieving land surface temperature from Landsat TM data and its application to the Israel-Egypt border region," *International Journal of Remote Sensing*, vol. 22, no. 18, pp. 3719–3746, Jan. 2001, doi: 10.1080/01431160010006971.
- [32] D. Sun and R. T. Pinker, "Estimation of land surface temperature from a Geostationary Operational Environmental Satellite (GOES-8)," *Journal of Geophysical Research: Atmospheres*, vol. 108, no. D11, Jun. 2003, doi: 10.1029/2002JD002422.
- [33] J. A. Sobrino, J. C. Jiménez-Muñoz, and L. Paolini, "Land surface temperature retrieval from LANDSAT TM 5," *Remote Sensing of Environment*, vol. 90, no. 4, pp. 434–440, Apr. 2004, doi: 10.1016/j.rse.2004.02.003.
- [34] G. Sòria and J. A. Sobrino, "ENVISAT/AATSR derived land surface temperature over a heterogeneous region," *Remote Sensing of Environment*, vol. 111, no. 4, pp. 409–422, Dec. 2007, doi: 10.1016/j.rse.2007.03.017.
- [35] F. Becker and Z. L. Li, "Surface temperature and emissivity at various scales: Definition, measurement and related problems," *Remote Sensing Reviews*, vol. 12, no. 3–4, pp. 225–253, Jan. 1995, doi: 10.1080/02757259509532286.
- [36] J. A. Sobrino, Z. L. Li, M. P. Stoll, and F. Becker, "Multi-channel and multi-angle algorithms for estimating sea and land surface temperature with ATSR data," *International Journal of Remote Sensing*, vol. 17, no. 11, pp. 2089–2114, Jul. 1996, doi: 10.1080/01431169608948760.
- [37] J. A. Sobrino, N. Raissouni, J. Simarro, F. Nerry, and F. Petitcolin, "Atmospheric water vapor content over land surfaces derived from the AVHRR data: application to the Iberian Peninsula," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1425–1434, May 1999, doi: 10.1109/36.763306.
- [38] J. A. Sobrino, V. Caselles, and F. Becker, "Significance of the remotely sensed thermal infrared measurements obtained over a citrus orchard," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 44, no. 6, pp. 343–354, Mar. 1990, doi: 10.1016/0924-2716(90)90077-O.
- [39] G. Zhou, R. Zhang, N. Liu, J. Huang, and X. Zhou, "On-Board Ortho-Rectification for Images Based on an FPGA," *Remote Sensing*, vol. 9, no. 9, 2017, doi: 10.3390/rs9090874.
- [40] W. G. Planet (ed), "Data extraction and calibration of TIROSN/NOAA radiometer," *NOAA Tech. Memo*, 1997.
- [41] NOAA, "NOAA Polar Orbiter Data Users Guide," NOAA, 1997. [Online]. Available: <https://www1.ncdc.noaa.gov/pub/data/satellite/publications/podguides/TIROS-N%20thru%20N-14/pdf/NCDCPD-ch1.pdf>