Research Article

A Review: Multiply and Accumulate Architectures for Digital signal Processing and digital image Processing

Kadi Jaya Ramesh [a], Chepuri Pravalika [b], Dr.Rajkumar Sarma [c]

[a] VLSI Design, School of Electronics and Electrical Engineering, Lovely Professional University, Punjab, India
[b] VLSI Design, School of Electronics and Electrical Engineering, Lovely Professional University, Punjab, India
[c] Assistant Professor, School of Electricals and Electronics Engineering, Faculty of Engineering and Technology, Jain(deemed-to-be-University), Ramanagar, Karnataka, India

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: The MAC (Multiply and Accumulate Unit) is the basic building block in the Digital signal processing and digital image processing systems. For efficient systems, the MAC unit should be fast with high precision and consuming low power. A MAC unit can be designed in Fixed-point arithmetic and Floating-point arithmetic. The use of Floating-point arithmetic gives high precision but it consumes more power and occupies more silicon area. To achieve high performance in a MAC unit, standard arithmetic can be implanted in its design. The IEEE-754 is floating-point arithmetic which can be used in the design of MAC. The use of the IEEE-754 standard improves the precision of the MAC unit. The conventional MAC units are designed by using HDL languages like the Verilog and the VHDL. By using these languages, the MAC unit can be designed in less time but designing the MAC unit at transistor level will increase the performance of the overall MAC unit. The cadence virtuoso can be used for designing the MAC at the transistor level. The design can be completed by using different technologies like 90nm gpdk or the tsmc 130nm etc. A floating-point MAC unit with half-precision using IEEE-754 in the transistor level will lead to better performance and high precision. Further we will discuss the different MAC architectures.

Keywords: MAC architectures, IEEE-754, Floating point, Digital signal Processing, Digital Image Processing

1. Introduction

In the Digital signal processing system, MAC (Multiplier and Accumulator) is the basic building block. The MAC consists of a multiplier and an accumulator, the multiplier multiplies the input samples and sends it to the accumulator which will add the present input and the past input and produce a new result that will be stored in a register [1][2][3].

Conventional MAC : Digital signal processing applications need high speed and low power MAC unit because the main operations in DSP like filtering and convolution are repetitively used [4][5]. MAC is also having applications in microprocessors and logic units [6]. So, A low power and highly efficient MAC is always recommended in the Digital signal processing applications [7]. A conventional MAC unit is shown in figure 1.

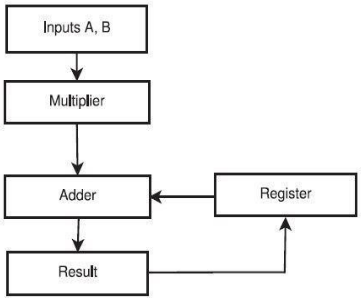


Figure.1.Showing Block diagram of Conventional MAC

MAC is mainly used in applications like filtering and convolution. In many DSP applications, a MAC with high precision is needed for better accuracy [8]. The overall performance of the DSP system can be enhanced by MAC with a floating-point architecture [9]. The use of a floating-point number of IEEE 754 format in a MAC results in a wider range of values at the cost of more storage and accuracy. More storage is needed in the case of floating-point because the position of the radix point needs to be encoded [10][11]. In the case of CPUs without the floating point architecture, a series of simple fixed-point architecture is used [12].

Types of MAC: A MAC can be implemented in fixed-point arithmetic and floating-point arithmetic [13][14].

(a) Fixed-point arithmetic MAC: In a fixed-point arithmetic MAC, the multiplier and accumulator used are of the fixed point. The use of fixed-point arithmetic results in less accuracy of output [15]. The block diagram of the fixed-point MAC is shown in figure 2.

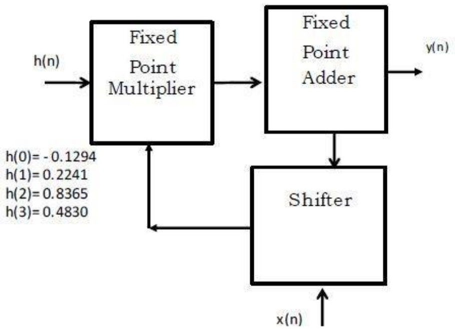


Figure.2.Showing Block diagram of Fixed-point MAC.

In this, the input samples are given to the fixed-point multiplier. The filter coefficients which are specified by the user and the multiplier is used to multiply filter coefficients and the input samples and the result is driven to the fixed-point adder. The fixed-point adder will add the present input to the previous result and the output is obtained as y(n) [12][16].

Limitations of fixed-point MAC:

1.ADC Quantization error.
2.Coefficient quantization error.
3.Overflow errors.
4.Round-off errors and truncation errors.

Floating-point arithmetic MAC: In a Floating-point arithmetic MAC, the multiplier and accumulator used are of the Floating point. The use of Floating-point arithmetic results in more accuracy of output. The block diagram of floating-point MAC is shown in figure 3.

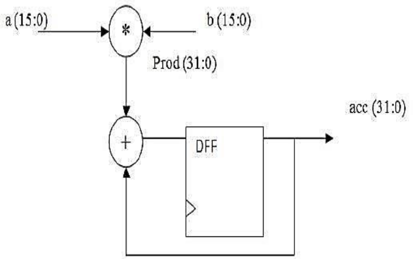


Figure.3.Showing Block diagram of Floating-point MAC.

In Floating-point MAC, the input samples are given to the multiplier. The multiplier will multiply the input samples and the result is passed to the floating-point adder. The adder will add the present input and the past result and produce a new output y(n) which is stored in a register [17]. The Floating-point MAC consumes more silicon area when compared to Fixed-point MAC but it will result in higher precision [18].

Floating-point Adder: The operation in a Floating-point adder is carried out in four steps.

1.Sorting.
2.Alignment.
3.Addition or Subtraction.
4.Normalization.

1.Sorting: In this step, the numbers are sorted in the decrement order i.e., from the largest number to the smallest number.

2.Alignment: In this step, the alignment of numbers is done to have the same exponent. This process is carried out by adjusting the exponent of a small number until the exponents of both numbers are matched.

3.Addition or Subtraction: In this step, the significands of the aligned numbers are added or subtracted.

4.Normalization: In this step, the result obtained will be normalized [19]. The block diagram of the Floating point addition is given in figure 4.

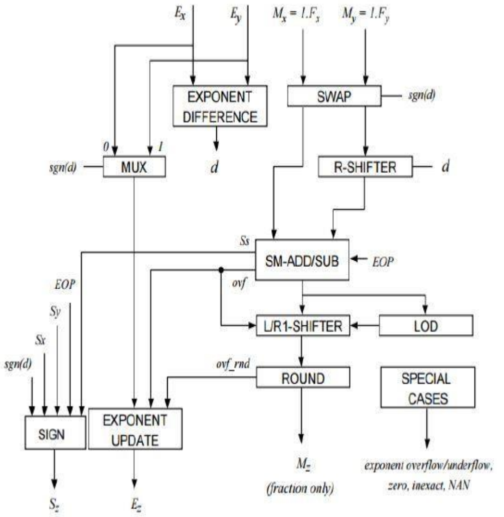


Figure.4.Showing Block diagram of floating-point addition

Floating-point Multiplier: Floating-point multipliers play an important role in Digital signal processing and digital image processing systems [20]. The block diagram of the Floating-point multiplier is given in figure 5.

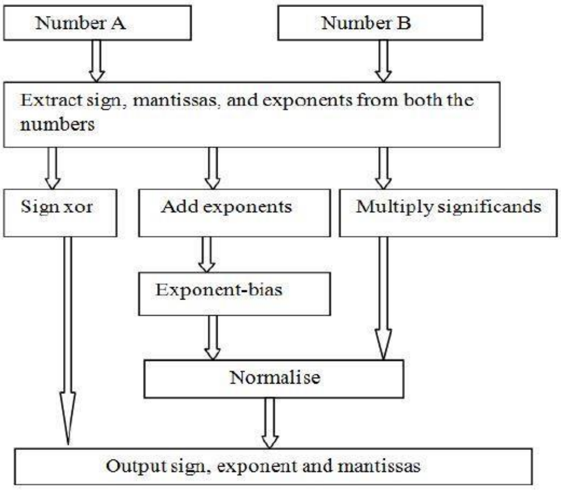

Number A
Number B
Extract sign, mantissas, and exponents from both the numbers
Sign xor
Add exponents
Multiply significands
Exponent-bias
Normalise
Output sign, exponent and mantissas

Figure.5.Showing Floating-point multiplier.

Let us assume the two floating-point numbers are m1 and m1 and the result obtained after multiplication is m, then

$$
\begin{aligned}
m &= m1*m2 \\
&= (-1)^{s1}.m1.2^{e1} * (-1)^{s2}.m2.2^{e2} \\
&= (-1)s1+s2.p1.p2.2e1+e2
\end{aligned}
$$

Standard IEEE 754 format: In 1985, the IEEE has released the standard binary Floating-point format. It includes different types of floating-point formats which include single precision and double precision, round mechanisms, arithmetic operations, etc [19]. The floating-point number can be represented by the given equation.

$$Z= (-1s) *2 (exp-bias)*(1*M)$$

According to the IEEE 754 format, a 32-bit floating point.

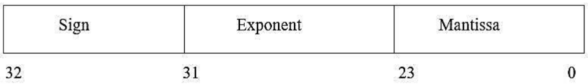
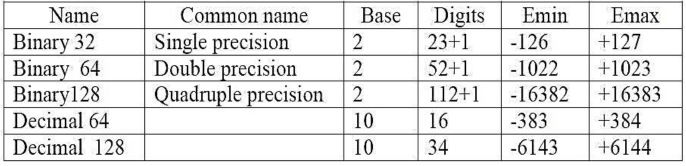


Figure.6.Showing IEEE 754 single-precision Floating-point format.

The standards which are defined in the IEEE 754 format are shown in table 1.Number consists of 8 bits which represent the exponential part, 23 bits represent the significant and the sign is represented by one bit [20].

Table.1.Showing Different standards in IEEE 754 format



Name	Common name	Base	Digits	Emin	Emax
Binary 32	Single precision	2	23+1	-126	+127
Binary 64	Double precision	2	52+1	-1022	+1023
Binary128	Quadruple precision	2	112+1	-16382	+16383
Decimal 64		10	16	-383	+384
Decimal 128		10	34	-6143	+6144

The first three formats are used for Binary floating point numbers and use 32,64 and 128 bits respectively. The last two formats are used for Decimal Binary floating-point numbers and use 64 and 128 bits respectively [10].

2. MAC Architectures:

1.A high throughput and low power dissipation MAC architecture is designed for DSP applications using the block enabling technique. The figure represents the pipeline MAC architecture with block enable technique.

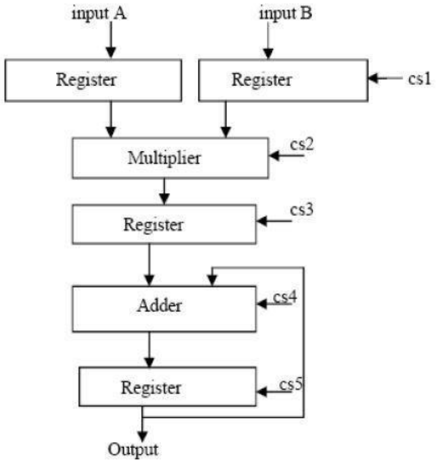


Figure.7.Showing Pipeline MAC architecture with block enable technique

The inputs to the MAC are fed to the multiplier block by using the input registers in a single clock cycle [16]. The delay of each block is found out and triggered at same amount of delay. The blocks will be enabled only when the inputs are available. The designed MAC unit results in a power-delay product of $177.95*10^{-15}$ J [2].

2.The reversible logic gates which are the fundamental building blocks of the Quantum computing are used for the design the MAC along with the vedic multiplier. The main advantage of the vedic multiplier is its power of calculating partial products in a single step [21]. If each value in a input can be mapped with unique value in output, then it is called a reversible logic Boolean function and the circuit with only reversible logic gates does not dissipate power [22].

`

This combination of vedic multiplier and the reversible logic in the design of MAC results in a efficient design. The criss cross method in Urdhava sutra is used to produce partial products. The increase in partial products will lead to critical paths in the design. The reversible logic gates which is used in optical and nano fields are used for reduction in the power reduction.

The MAC architecture is simulated with various multipliers like Booth, Wallace and vedic multiplier. The obtained power, area and speed parameters with different architectures is given in table 2 [23].

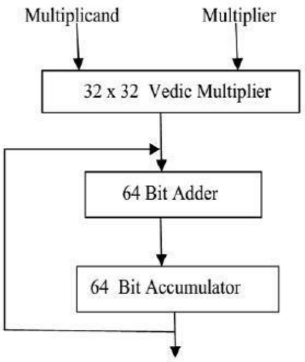
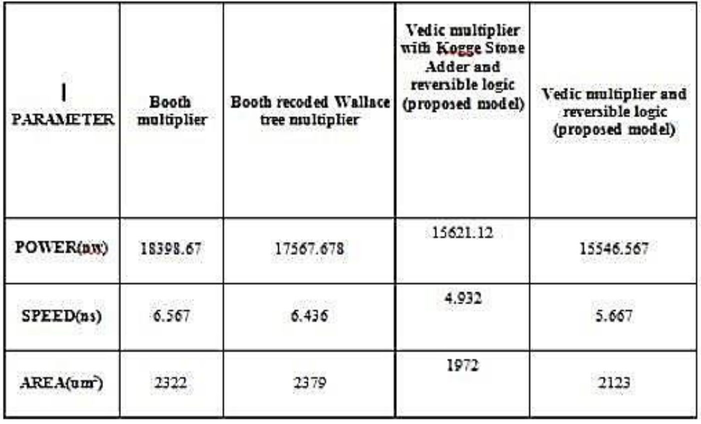


Figure.8.Showing Modified MAC Architecture.

Table.2.Showing Analysis of MAC with different multipliers.



| PARAMETER	Booth multiplier	Booth recoded Wallace tree multiplier	Vedic multiplier with Kogge Stone Adder and reversible logic (proposed model)	Vedic multiplier and reversible logic (proposed model)
POWER(nw)	18398.67	17567.678	15621.12	15546.567
SPEED(ns)	6.567	6.436	4.932	5.667
AREA(um^{2})	2322	2379	1972	2123

3. A 2 cycle multiply accumulate architecture is proposed which uses guarding bits to efficiently run on longer MAC loops. In this architecture, the final result requires no extra cycles because the carry propagation is done in the second stage of the MAC pipeline. This architecture produces sum and the accumulated value in each cycle.

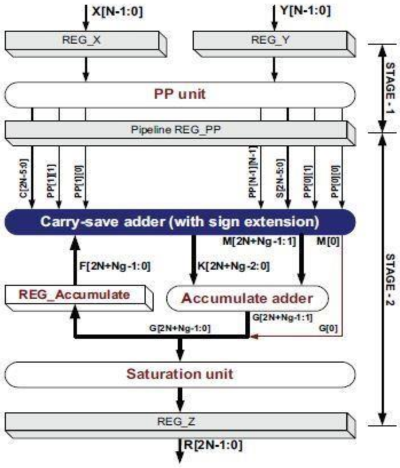


Figure.9.Showing Block diagram of 2 cycle MAC Architecture.

A carry save adder is added after the pipeline registers whose delay is equal to that of a single full adder. The final adder is removed and the carry save adder is used inside the accumulate stage. This architecture is proved to be energy efficient than the conventional 2 cycle MAC architecture [24].

4. A low power compressor architecture is proposed which is included in the partial product reduction stage which reduces the power consumption of the design. The presence of dedicated MAC units in filters results in faster FFT and FIR computations [25].

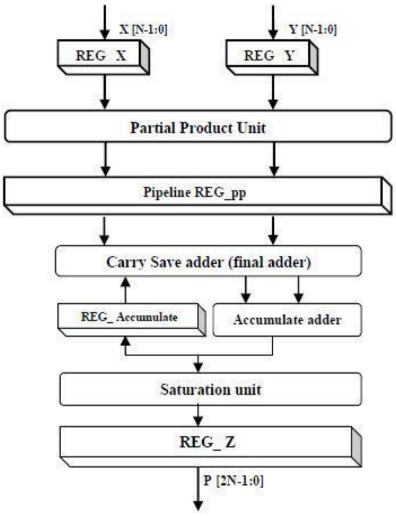
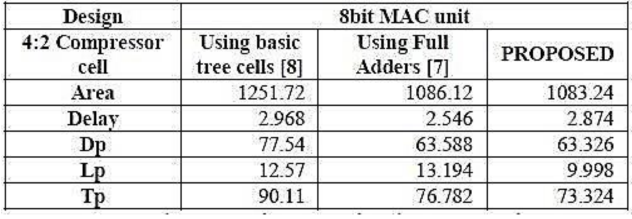


Figure.10.Showing Block diagram of MAC with compressor architecture in partial product reduction stage.

A large number of compressors are used in the design using the full adders which increases the speed of multiplication [26]. Since the full adders are used, the signal drive strength required is more which costs more power.

The designed is simulated in the cadence RTL with TSMC 65nm node and the results are compared with conventional MAC architecture in the table 3 [27].

Table.3.Comparison of conventional MAC and the proposed MAC in ASIC domain.



Design	8bit MAC unit		
4:2 Compressor cell	Using basic tree cells [8]	Using Full Adders [7]	PROPOSED
Area	1251.72	1086.12	1083.24
Delay	2.968	2.546	2.874
Dp	77.54	63.588	63.326
Lp	12.57	13.194	9.998
Tp	90.11	76.782	73.324

5. A low power and highly efficient floating-point MAC architecture is proposed which uses BCD blocks and the output of the MAC is also produced in the BCD format.

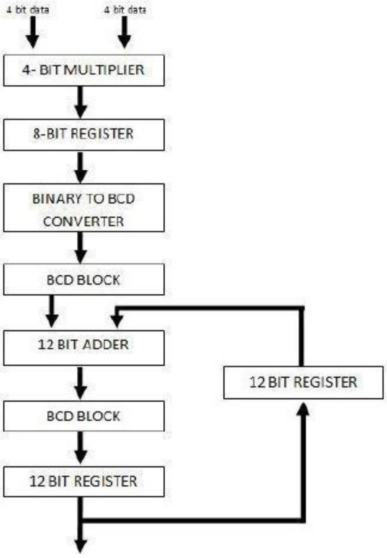


Figure.11.Showing Proposed MAC architecture with BCD block.

Here in this architecture, a binary to BCD converter and BCD block are used. The binary to BCD converter is used because the output of the multiplier is in binary format which need to be sent to the BCD block. An efficient MAC architecture is designed and is analyzed using cadence virtuoso in 90nm technology. The power, delay and area analysis are done for each block which yields efficient results [28].

3. Conclusion

This paper discusses about the fundamentals of the Multiply and accumulate unit which is the most important part in a Digital signal processing. The different types of MAC architectures are reviews and their effectiveness is discussed along. The architecture that is developed from transistor level shows a significant better result than that of the architectures that are designed in HDL. The use of standard format like IEEE 754 results in a better performance in the design.

References

1. Shanthala S, Kulkarni. S.Y.,"VLSI Design and Implementation of Low Power MAC Unit with Block Enabling Technique," European Journal of Scientific Research ISSN 1450-216X.
2. Sen, Avisek, Partha Mitra, and Debarshi Datta. "Low power mac unit for DSP processor." International Journal of Recent Technology and Engineering (IJRTE) 1.6 (2013): 93-95.
3. Israel Koren, Computer Arithmetic Algorithms, A K Peters, second edition, 2002.
4. J. J. F. Cavanagh, Digital Computer Arithmetic. New York: McGraw-Hill, 1984.
5. Rakesh, S., and KS Vijula Grace. "A survey on the design and performance of various MAC unit architectures." 2017 IEEE International Conference on Circuits and Systems (ICCS). IEEE, 2017.
6. A. Abdelgawad, Magdy Bayoumi ," High Speed and AreaEfficient Multiply Accumulate (MAC) Unit for Digital Signal Processing Applications", IEEE Int. Symp. Circuits Syst. (2007) 3199–3202.
7. S. J. Jou, C. Y. Chen, E. C. Yang and C.C.Su, "A pipeline Multiplier-Accumulator using a high speed, low power static and dynamic full adder design", IEEE custom Integrated circuit conference, 1995, pp. 593-596.
8. Saravanan, R., P. Balaji, and R. Prabu. "Design of 16-bit floating point multiply and accumulate unit." IJMTES Int. J. Mod. Trends Eng. Sci 3.01 (2015).
9. Mehta, Sonali, Balwinder Singh, and Dilip Kumar. "Performance Analysis of Floating Point MAC Unit." International Journal of Computer Applications 78.1 (2013).
10. Jyoti Singh Chouhan, Nitin Jain. "Fused floating point mac (multiply and add) unit with configurable architecture" Journal of Scientific Research in Allied Sciences. ISSN No. 2455-5800 (2016).
11. L. A. Tawalbeh, "Radix-4 ASIC Design of a Scalable Montgomery Modular Multiplier using Encoding Techniques," M.S. Thesis, Oregon State University, USA, October 2002.
12. John L. Hennessy and David A. Patterson. Computer Architecture A Quantitative Approach, Second Edition. Morgan Kaufmann, 1996.
13. Deepika Setia, Charu Madhu, "Novel Architecture of High Speed Parallel MAC using Carry Select Adder", nternational Journal of Computer Applications (0975 – 8887) Volume 74– No.1, July 2013.
14. N. J. Babu and R. Sarma, "A novel low power and high speed Multiply-accumulate (MAC) unit design for floating-point numbers," 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Chennai,2015, pp. 411-417, doi: 10.1109/ICSTM.2015.7225452.
15. M. J. Flynn, S. F. Oberman, AdvancedComputer Arithmatic Design. John Wiley & Sons, Inc, 2001.
16. West and Harris, CMOS VLSI Design: a circuits and systems perspective, Addison-Wesley Publishing Company,3rd ed
17. R. P. Rao, N. D. Rao, K. Naveen and P. Ramya, "implementation of the standard floating-point mac using ieee 754 floating point adder", 2018 Second International Conference on Computing Methodologies and Communication (ICCMC), Erode, 2018, pp. 717-722.
18. Saokar, Sandesh S., R. M. Banakar, and Saroja Siddamal. "High speed signed multiplier for digital signal processing applications." 2012 IEEE International Conference on Signal Processing, Computing and Control. IEEE, 2012.
19. A. Goldovsky and et al., "Design and Implementation of a 16 by 16 Low-Power Two's Complement Multiplier". In IEEE International Symposium on Circuits and Systems, 5, pp 345–348, 2000.
20. Al-Ashrafy, Mohamed, Ashraf Salem, and Wagdy Anis. "An efficient implementation of floating point multiplier." 2011 Saudi International Electronics, Communications and Photonics Conference (SIECPC). IEEE, 2011.
21. R.Bhaskar, Ganapathi Hegde, P.R.Vaya," An efficient hardware model for RSA Encryption system using Vedic mathematics", International Conference on Communication Technology and System Design 2011 Procedia Engineering 30 (2012) 124 – 128.
22. C.H. Bennett," Logical reversibility of computation", IBM J.Res. Dev. 17 (1973) 525–532.
23. Anitha, R., et al. "A 32 bit mac unit design using vedic multiplier and reversible logic gate." 2015 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]. IEEE, 2015.

24. Hoang, Tung Thanh, Magnus Själander, and Per LarssonEdefors. "High-speed, energy-efficient 2-cycle multiply-accumulate architecture." 2009 IEEE International SOC Conference (SOCC). IEEE, 2009.
25. Tung Thanh Hoang; Sjalander, M.; Larsson-Edefors, P., "AHigh-Speed, Energy-Efficient Two-Cycle Multiply-Accumulate (MAC) Architecture and Its Application to a Double-Throughput MAC Unit," Circuits and Systems I:Regular Papers, IEEE Transactions on , vol.57, no.12,pp.3073,3081, Dec. 2010.
26. Kiwon Choi; Minkyu Song, "Design of a high performance 32×32-bit multiplier with a novel sign select Booth encoder," Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on, vol.2, no.,pp.701,704 vol. 2, 6-9 May 2001.
27. Narendra, C. P., and KM Ravi Kumar. "Low power MAC architecture for DSP applications." International Conference on Circuits, Communication, Control and Computing. IEEE, 2014.
28. Babu, N. Jithendra, and Rajkumar Sarma. "A novel low power and high speed Multiply-accumulate (MAC) unit design for floating-point numbers." 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM). IEEE, 2015.