IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 4 , 2022**

Efficient Hardware Implementation of 2D-DCT Architecture for Versatile Video Compression

Srikanth Reddy P[1] , Chella Santhosh[2]

Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation (KLEF), Vaddeswaram, Green fields, Guntur, Andhra Pradesh, India -522302

DOI : 10.48047/IJFANS/11/ISS4/112

Abstract

Versatile Video Coding is a new international image compression standard developed to provide high compression factor compared to previous standards like H.E.V.C,H,264 . VVC uses various DCT algorithms for image compression & IDCT algorithms for reconstruction, VVC provides better compression factor at expense of high computational complexity. This paper presents an approach for hardware implementation of 8X8 DCT,IDCT modules in Design 1, in which a 512x512 pixel image is fed as input to the DCT module and it is retrieved at IDCT.As VVC implementation utilizes hardware resources at higher expense this paper mainly focuses on reconstruction off image, reduction of resources especially multipliers .In order to reduce multipliers count a methodology has been implemented in Design 2[1].To eliminate multipliers completely a methodology has been implemented in Design 3 through adding and shifting operations. In order to reduce adders and shifters count a methodology has been proposed and implemented in Design 4. The proposed methodology reduces 4 stages of shifting and adding operations to 2 stages with complete elimination of multipliers. All the Designs implemented in this paper can generate 64 Transformed coefficient per Cycle [2]. The design implemented in proposed methodology can be utilized in high performance area efficient low power VVC modules. All the four Designs are simulated and synthesized in Xilinx Vivado 2019.2 shows that the simulation results are same for all the implemented Designs[3]. However Synthesis reports shows that the Proposed methodology is ideal as the hardware utilization is low which has been implemented on FPGA (Zed Board).

Keywords:Versatile Video Coding(V.V.C),Discrete Cosine Transform(D.C.T), Inverse Discrete Cosine Transform(I.D.C.T).

1. Introduction

In order to reduce the utilization of transmission band width, hardware disc space and higher throughput image compression technique is utilized. Lossy image compression technique is preferred by JPEG for multimedia transmissions where data loss up to considerable amount can be tolerated. DCT is a lossy image compression algorithm which is utilized in HEVC,VVC compression standards[5]. After the application of DCT algorithm on the image the obtained values are quantized and transmitted. At the receiver end the obtained values are dequantized, IDCT algorithm is implemented for reconstruction of image. VVC deals with Transform Units (TU) of size

4x4,8X8,16X16,32X32,64X64 for better compression and faster processing rate where as HEVC deals with up to 32X32.As TU size increases computational complexity, Hardware utilization increases.

DCT is a lossy image compression algorithm which can be implemented in the form of matrix on the input image which is also taken in the same form. Similarly, IDCT is applied for data retrieval [5].

2- Dimension DCT Equation

For a N X N Image u(m,n), Forward DCT is defined as

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 4 , 2022**

$$\vartheta(k,l) = \alpha(k)\alpha(l)\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} u(n)\cos\frac{\pi(2n+1)k}{2N}\cos\frac{\pi(2n+1)l}{2N}$$

$$0 \le k \le N-1$$

$$1 \le k \le N-1$$

Where $\alpha(0) = \frac{1}{\sqrt{N}}$ & $\alpha(k) = \frac{2}{\sqrt{N}}$

Inverse DCT (IDCT) is

$$u(m,n) = u(k) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} \alpha(k)\alpha(l)\vartheta(k,l)\cos\frac{\pi(2n+1)k}{2N}\cos\frac{\pi(2n+1)}{2N}$$

$$0 \le m,n \le N-1$$

Coefficient matrix

N*N cosine transform matrix is given by C = {c(k,n)}

$$C(k,n) = \begin{cases} \frac{1}{\sqrt{N}}, & k = 0, 0 \le n \le N-1 \\ \sqrt{\frac{2}{N}}\cos\frac{\pi(2n+1)k}{2N}, & 1 \le k \le N-1 \\ & 0 \le n \le N-1 \end{cases}$$

In this paper 3 designs are implemented In Design 1 8X8 DCT is implemented on a 512X512 image and the results obtained are fed as input for 8X8 IDCT. The results obtained are reconverted to image form. As 8X8 DCT, IDCT are implemented it deals with higher computational complexity and has large number of multipliers [6]. In hardware implementation multipliers have significant importance as they occupy more area. It is a specification to eliminate or to reduce them as much as possible [7].

Design 2 is designed to reduce the multipliers count. As DCT involves two stages of multiplication for 2 dimensions, similarly IDCT has another 2 stages Design 1 deals with 4 stages of multiplications. In Design 2, two multiplexers are used to reduce the multiplication stages from 4 to 2[8]. However this method leads to increase the registers count as the outputs obtained in each stage is stored and assigned to another set of registers in transposed manner for further processing's. But the multiplier count is reduced which has higher significance in terms of hardware reduction. This method is applied for the same input 512X512 image.

Design 3 is designed to eliminate multipliers completely which can be achieved through adders/subtracters and shifters. Multiplication with Constant values is achieved through adding/subtracting and shifting operations. This method is similar to Design 1 except for the complete elimination of multipliers with adders and shifters. However it still deals with 4 stages of operation has larger number of adders and shifters.

For verification purpose image values stored in a particular location at input side which are fed to DCT are compared with the outputs at IDCT [6]. The results shows that they are almost equal. This observation has been made for all the four designs. After synthesis it has been observed that Multipliers generated in Design 2 are less in number compared to Design 1. Design 3 has zero (0) multipliers as it is implemented with adding and shifting operations [9]. Proposed methodology which is Design 3 has lesser number of adding and shifting operations compared to Design 2. With zero multipliers at expense of increased multiplexers and registers it is the idealized hardware model in the paper.

2. Design/Methods/Modelling

Design 1:

Top level representation of the VVC MODULE (upto DCT, IDCT implementation) is shown in figure 1

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 4 , 2022**

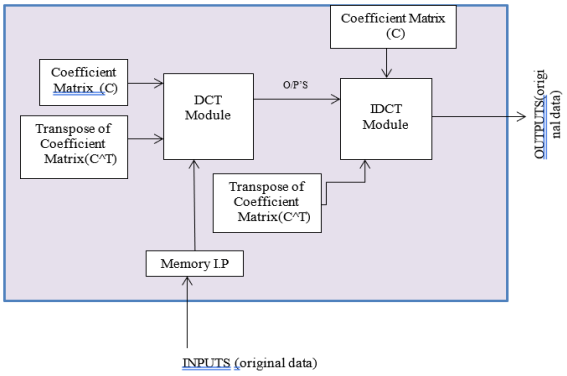


Fig.1.VVC module (DCT,IDCT level implementation)

8X8 DCT coefficients generated from algorithm are shown in form of variables for simplicity in equation 1.DCT, IDCT operations are performed using this coefficient matrix.

$$C = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & c & f & g & -g & -f & -c & -b \\ d & e & -e & -d & -d & -e & e & d \\ c & -g & -b & -f & f & b & g & -c \\ a & -a & -a & a & a & -a & -a & a \\ f & -b & g & c & -c & -g & b & -f \\ e & -d & d & -c & -e & d & -d & e \\ g & -f & c & -b & b & -c & f & -g \end{bmatrix}$$ -----eqn .1

Implemented architecture of the DCT module is shown in fig 2

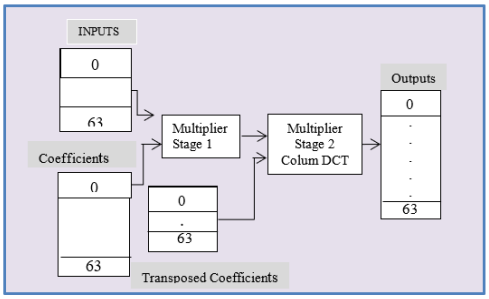


Fig.2. DCT Module

A 512X512 image shown in figure 3 is the input fed to the DCT module. A program has been developed in MATLAB to convert this image into pixel values. These pixel values generated for the entire image is stored in a COE FILE.

Fig.3: Input Image.[11]

A memory IP shown in figure 4 has been generated in VIVADO through Verilog. The COE file generated is loaded into this Memory IP. This memory is a simple dual port RAM consisting of 2,62,144 input memory locations, each location consisting of an 8-bit data (a pixel). Total number of bits for the entire image is 512X512X8=20,97,152 bits. At the other end of this dual

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 4 , 2022**

port RAM 512 bits are retrieved using a counter. These 512 bits are portioned and stored in 64 registers each of 8-bit width. These registers are fed as inputs to the DCT module. After processing 64 outputs have been produced from this 8x8 module per clock cycle. This process is repeated for entire input image that is for all the generated pixel values.

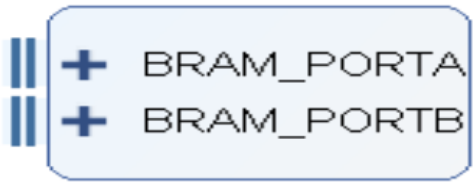


Fig. 4: Memory IP generated in DCT Module.

IDCT

The outputs obtained from DCT are fed to the IDCT module shown in figure 5. After intermediate and final processing, the design is simulated, and results obtained are stored in registers. Using counters entire operation has been controlled. A comparison has been made between inputs and final outputs with respect to address location (1397) and at particular instant of counter value that is at particular time period. This comparison shows that they are almost same with minor differences. These Differences are due to data loss during compression and reconstruction of image (as it is lossy image compression). The output values obtained for entire input image are printed in a text file. These values are converted to image form which is the reconstructed image shown in figure 13. The entire design is synthesized for the observation of resource utilization.

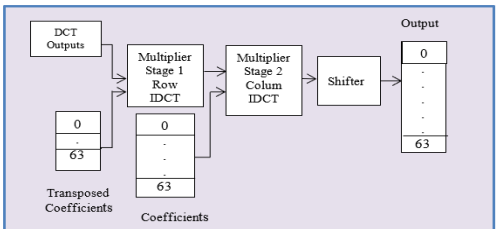


Fig.5: IDCT Module

Design 2:

This design and implementation is similar to Design 1 except it uses two multiplexers one at DCT and other at IDCT in order to reduce four stages of multiplication operations to two stages of operations which reduces the multipliers count from 1244 (Design 1) to 744(Design 2).However the register count is increased as 64 more registers are used to store intermediate output values which are fed in transposed manner as input to multiplexer for further processing[10].

The end results are verified and observed that they are similar to Design1.Block level representation of implemented architectures of Optimized DCT, IDCT modules are shown in figures 6,7.

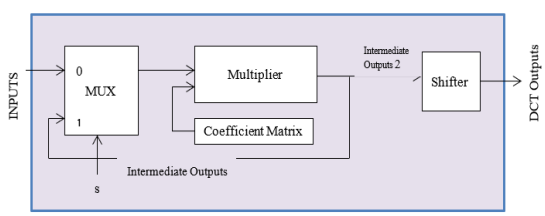


Fig.6:Optimized DCT Module

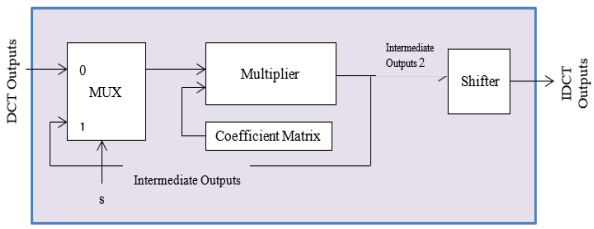


Fig.7: Optimized IDCT Module

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 4 , 2022**

Design 3:

It is a specification in hardware implementation to reduce or to eliminate multipliers completely in order to minimize area occupation and power utilization.This can be achieved through adding/subtracting and shifting operations. The DCT, IDCT coefficients are always constant in the design.so the input values stored in registers are added/subtracted and shifted according to coefficients values. This process has been implemented for both DCT, IDCT modules. This process completely eliminates multipliers but as it is similar to Design 1 which involves 4 stages of operations (2 for DCT,2 for IDCT) leads to utilization of adders, shifters in large number. In this implementation adding/subtracting and shifting operations has been done on 256 coefficients. However this methodology completely eliminates multipliers. Total multiplier count is null (0). The entire process of implementation is same as Design 1.

Implemented Architecture of Multiplier less DCT, IDCT modules are shown in figures 8,9.

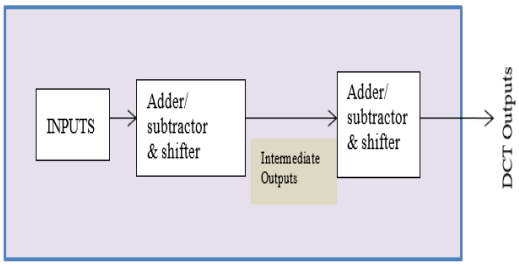


Fig.8:Multiplierless DCT module

Provide sufficient detail methods to allow the work to be reproduced. Methods already published should be indicated by a reference: only relevant modifications should be described.

3. Results and Discussion

Output image generated after implementation of DCT, IDCT is shown in figure 9 which is the reconstructed image

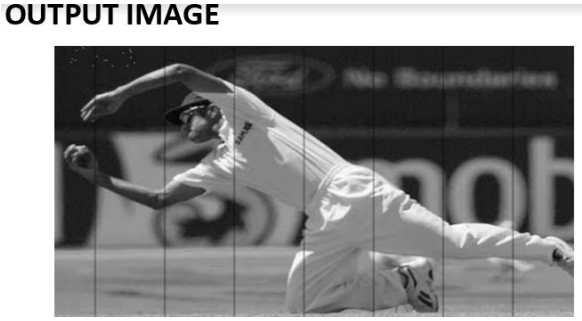


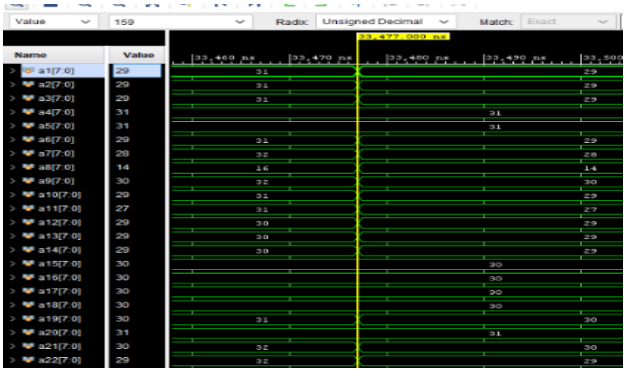
Fig.9: Output image

The input image fed to the DCT module is retrieved at the IDCT module with some distortions. As it is a lossy image compression technique there is some degradation in the image quality. However in real time processing applications smoothening algorithms and image restoration filters are used to overcome these issues.

For verification purpose a 8X8 chunk of input data at a particular input memory location (1397) is verified with the output chunk of 8X8 data at that instant (with respect to input location) for all the four Designs. For simple representation the values generated for Design 3 are shown in Figures 10,11.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 4 , 2022**

Inputs at address location 1397 are snipped from the XILINX VIVADO tool are shown in figure 10.



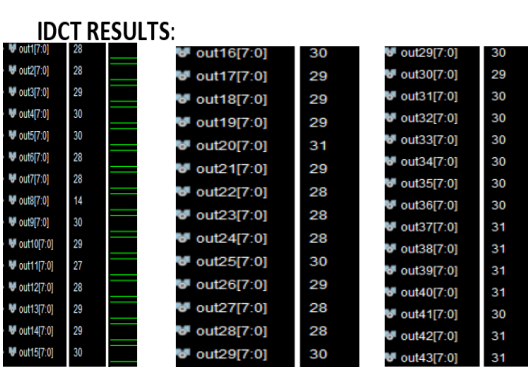


Fig.10: Input values

Fig.11: IDCT Values from Vivado

Generated output values at the IDCT Module are shown in figure 11.

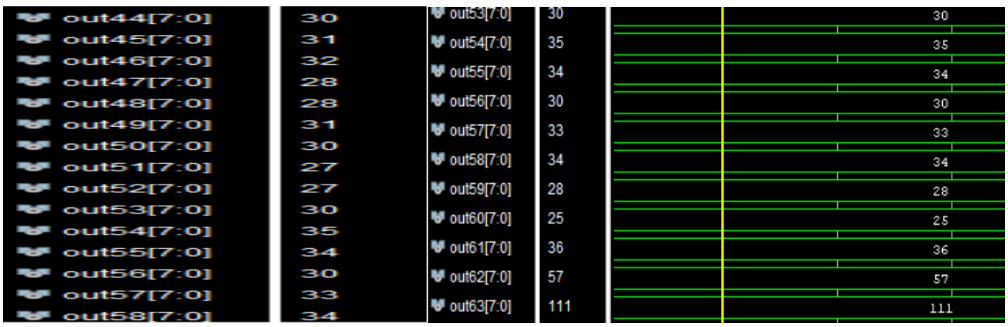


Fig.12:Outputs

Synthesis Results

MODULE	DESIGN 1		DESIGN 2	
	MULTIPLIERS	ADDERS	MULTIPLIERS	ADDERS
DCT	588	64	416	65
IDCT	656	128	328	65
TOTAL	1244	192	744	130

DESIGN 3

MODULE	ADDERS	Total number of utilized coefficients	REGISTERS	MULTIPLIERS
IDCT	496	128	184	0
DCT	360	128	200	0
TOTAL	856	256	384	0

VIVADO SYNTHESIS REPORT

ISSN PRINT 2319 1775 Online 2320 7876

Research paper Volume 11, Iss 4 , 2022

1.Generated for Design1(DCT,IDCT) is shown in Figure 12

```
Module dc
Detailed RTL Component Info :
+---Adders :
      24 Input     46 Bit       Adders := 24
      20 Input     46 Bit       Adders := 8
       8 Input     46 Bit       Adders := 8
      19 Input     46 Bit       Adders := 8
      18 Input     46 Bit       Adders := 8
       2 Input     26 Bit       Adders := 64
      25 Input     26 Bit       Adders := 32
       9 Input     26 Bit       Adders := 8
      21 Input     26 Bit       Adders := 16
       8 Input     26 Bit       Adders := 8
       7 Input     16 Bit       Adders := 8
       3 Input     15 Bit       Adders := 8
       2 Input     14 Bit       Adders := 64
       5 Input     14 Bit       Adders := 8
       3 Input     14 Bit       Adders := 8
       2 Input     13 Bit       Adders := 40
       2 Input     12 Bit       Adders := 40
+---Registers :
                   46 Bit    Registers := 56
                   26 Bit    Registers := 64
                   16 Bit    Registers := 8
```

```
Module dct_1
Detailed RTL Component Info :
+---Adders :
         8 Input     41 Bit         Adders := 64
         2 Input      2 Bit         Adders := 1
+---Registers :
                     41 Bit       Registers := 128
                      2 Bit       Registers := 1
                      1 Bit       Registers := 1
+---Multipliers :
                       8x41   Multipliers := 192
                       7x41   Multipliers := 160
                       6x41   Multipliers := 64
Module idct
Detailed RTL Component Info :
+---Adders :
         8 Input     81 Bit         Adders := 64
         2 Input      3 Bit         Adders := 1
+---Registers :
                     31 Bit       Registers := 128
                      8 Bit       Registers := 64
                      3 Bit       Registers := 1
                      1 Bit       Registers := 1
+---Multipliers :
                       8x31   Multipliers := 160
                       7x31   Multipliers := 104
                       6x31   Multipliers := 64
```

Fig.12: Design 1 report Fig.13: Design 2 report

Generated for Design 2 is shown in Figure 13

```
Module dc
Detailed RTL Component Info :
+---Adders :
      24 Input     46 Bit       Adders := 24
      20 Input     46 Bit       Adders := 8
       8 Input     46 Bit       Adders := 8
      19 Input     46 Bit       Adders := 8
      18 Input     46 Bit       Adders := 8
       2 Input     26 Bit       Adders := 64
      25 Input     26 Bit       Adders := 32
       9 Input     26 Bit       Adders := 8
      21 Input     26 Bit       Adders := 16
       8 Input     26 Bit       Adders := 8
       7 Input     16 Bit       Adders := 8
       3 Input     15 Bit       Adders := 8
       2 Input     14 Bit       Adders := 64
       5 Input     14 Bit       Adders := 8
       3 Input     14 Bit       Adders := 8
       2 Input     13 Bit       Adders := 40
       2 Input     12 Bit       Adders := 40
+---Registers :
                   46 Bit    Registers := 56
                   26 Bit    Registers := 64
                   16 Bit    Registers := 8
```

```
Module d
Detailed RTL Component Info :
+---Adders :
         2 Input     31 Bit         Adders := 64
        25 Input     31 Bit         Adders := 32
        21 Input     31 Bit         Adders := 16
         9 Input     31 Bit         Adders := 8
         8 Input     31 Bit         Adders := 8
         2 Input     13 Bit         Adders := 1
         2 Input      3 Bit         Adders := 1
         2 Input      2 Bit         Adders := 1
+---Registers :
                    512 Bit       Registers := 1
                     31 Bit       Registers := 128
                     13 Bit       Registers := 1
                      8 Bit       Registers := 64
                      3 Bit       Registers := 1
                      1 Bit       Registers := 1
```

. Generated for Design 3 is shown in Figure 14

4. Conclusions

A 512X512 image is converted to pixel values is fed as input to DCT module in Design 1 whose outputs are fed to IDCT module. The IDCT outputs are reconstructed into image form This Process has been implemented for all the Designs. In order to reduce multiplier, count a methodology has been implemented in Design2 using multiplexers, In order to eliminate multipliers completely a methodology has been implemented in Design 3 through Adders and Shifters. To reduce adders and transform coefficients to be shifted a methodology has been proposed and implemented in Design 4. This proposed module has been developed on Zed Board and outputs are observed.

5.References

[1] -ShubhLakshmiAgrwal, DeekshaKumari,MeetaSharma,Sandeepkumar Gupta, Improved Image Compression Technique Using IWT-DCT Transformation 2016 2nd International Conference on Next Generation Computing Technologies (NGCT-2016) Dehradun, India 14-16 October2016

[2] Mohamed, B., Elsayed, A., Amin, O., Khafagy, E., Abdelrasoul, M., Shalaby, A., & Sayed, M. S. (2017, December). High-level synthesis hardware implementation and verification of HEVC DCT on SoC-FPGA. In 2017 13th International Computer Engineering Conference (ICENCO) (pp. 361-365). IEEE.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 4 , 2022**

[3] Jridi, M., & Meher, P. K. (2016). Scalable approximate DCT architectures for efficient HEVC-compliant video coding. IEEE Transactions on Circuits and Systems for Video Technology, 27(8), 1815-1825.

[4] Mert, A. C., Kalali, E., & Hamzaoglu, I. (2017). High performance 2D transform hardware for future video coding. IEEE Transactions on Consumer Electronics, 63(2), 117-125.

[5] Chidanandan, A., Moder, J., & Bayoumi, M. (2006, August). Implementation of NEDA-based DCT architecture using even-odd decomposition of the 8× 8 DCT matrix. In 2006 49th IEEE International Midwest Symposium on Circuits and Systems (Vol. 1, pp. 600-603). IEEE.

[6] Yu, S., & Swartziander, E. E. (2001). DCT implementation with distributed arithmetic. IEEE transactions on Computers, 50(9), 985-991.

[7] K. Kim, S. Jang, S. Kwon, and K. Son, ªAn Improvement of VLSI Architecture for2-Dimensional Discrete Cosine Transform and Its Inverse,º Proc. SPIE, vol. 2727, pp. 1017-1026, 1996.

[8] Mannepalli, K., Sastry, P. N., & Suman, M. (2017). Accent recognition system using deep belief networks for telugu speech signals. In *Advances in Intelligent Systems and Computing* (Vol. 515, pp. 99–105). Springer Verlag

[9] Mannepalli, K., Sastry, P. N., & Suman, M. (2016). FDBN: Design and development of Fractional Deep Belief Networks for speaker emotion recognition. International Journal of Speech Technology, 19(4), 779–790. https://doi.org/10.1007/s10772-016-9368-y

[10] Urruela-Barrios R, Ramírez-Cedillo E, Díaz de León A, Alvarez A, Ortega-Lara W (2019) Alginate/gelatin hydrogels reinforced with TiO2 and β-TCP rabricated by microextrusion-based printing for tissue regeneration. Polymers 11(3):457