

## FPGA-Based Design and Implementation of 16-Point FFT Calculator

Nguyen Van Thanh Loc, Hoang Truong Huu Thuong, Mai Thi Hoai Nhi, Do Duy Tan\*

Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam

\* Corresponding author. Email: [tandd@hcmute.edu.vn](mailto:tandd@hcmute.edu.vn)

### ARTICLE INFO

Received: 17/2/2022  
Revised: 18/5/2022  
Accepted: 1/8/2022  
Published: 30/8/2022

### KEYWORDS

16-point FFT;  
FPGA;  
R2SDF;  
Pipeline;  
Testbench.

### ABSTRACT

This paper presents a design and construction of a 16-point FFT calculator based on FPGA technology. Specifically, the data is complex numbers in which the real and imaginary parts are represented as fixed-point real numbers. Moreover, a fixed-point real number is defined by 16 bits with the high-significant bit being the 2's complement bit, the next 9 bits being the integer part, and the last 6 bits being the fractional part. By means of simulations and FPGA board-based experimental results, we show the advantage of the proposed design compared to the existing ones. The operating frequency of the system is 149,867 MHz giving 4,683,343 FFT calculations of 16-points per second with low error (only about 0.3). This design of FFT calculator could be extensible to perform multi-point transformations since it is designed in a pipeline architecture with modules that are easily resizable and can be embedded in systems that require the 16-point FFT calculator.

## Thiết Kế Và Thi Công Bộ Tính Toán FFT 16 Điểm Dựa Trên Công Nghệ FPGA

Nguyễn Văn Thành Lộc, Hoàng Trương Hữu Thương, Mai Thị Hoài Nhi, Đỗ Duy Tân\*

Khoa Điện-Điện Tử, Trường Đại Học Sư Phạm Kỹ Thuật TP HCM, Việt Nam

\* Tác giả liên hệ. Email: [tandd@hcmute.edu.vn](mailto:tandd@hcmute.edu.vn)

### THÔNG TIN BÀI BÁO

Ngày nhận bài: 17/2/2022  
Ngày hoàn thiện: 18/5/2022  
Ngày chấp nhận đăng: 1/8/2022  
Ngày đăng: 30/8/2022

### TỪ KHÓA

16-point FFT;  
FPGA;  
R2SDF;  
Pipeline;  
Testbench.

### TÓM TẮT

Bài báo này trình bày thiết kế và thi công một bộ tính toán FFT (Fast Fourier transform) 16 điểm dựa trên công nghệ FPGA với kiểu dữ liệu là số phức trong đó phần thực và phần ảo được biểu diễn dưới dạng số thực dấu chấm tĩnh. Trong đó, một số thực dấu chấm tĩnh được xác định bằng 16 bit với bit trọng số cao là bit dấu bù 2, 9 bit tiếp theo là phần nguyên, 6 bit cuối là phần phân số. Thông qua các kết quả đánh giá qua mô phỏng và thực thi thực tế trên kit FPGA, chúng tôi chỉ ra tính hiệu quả của thiết kế được đề xuất so với một số thiết kế đang có. Tần số hoạt động của hệ thống là 149.867 MHz cho ra 4,683,343 phép tính FFT 16 điểm mỗi giây và sai số của các kết quả thấp (chỉ khoảng 0.3). Từ bộ tính toán FFT này, có thể mở rộng để thực hiện các biến đổi nhiều điểm hơn do được thiết kế theo kiến trúc pipeline với các khối dễ dàng thay đổi kích thước cũng như có thể nhúng vào các hệ thống yêu cầu bộ tính toán FFT 16 điểm.

Doi: <https://doi.org/10.54644/jte.71B.2022.1139>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

### 1. Giới thiệu

Biến đổi Fourier nhanh (Fast Fourier transform - FFT) là một thuật toán cốt lõi, được sử dụng rộng rãi trong các hệ thống xử lý tín hiệu số và truyền thông. Thuật toán FFT được triển khai đầu tiên vào năm 1965 bởi Cooley và Tukey [1] và phát triển thành nhiều kiểu kiến trúc với mong muốn tối ưu về

thời gian tính toán, tài nguyên phần cứng, năng lượng tiêu hao. Hiện nay, kích thước bộ tính toán FFT ngày càng lớn như từ 64 điểm với IEEE 802.11a [2] cho đến 256/512 điểm với IEEE 802.11ac [3] và mới nhất hiện nay là 256-2048 điểm cho IEEE 802.11ax [4]. Ngoài ra hệ thống 4G LTE cũng yêu cầu bộ FFT có kích thước 128-2048 điểm [5].

Từ [6] có thể thấy rằng các kiến trúc pipeline FFT được xem xét để thực thi có nhược điểm là phải tăng số tầng thiết kế khi số điểm FFT tăng dẫn đến tốc độ tính toán giảm. Để khắc phục điều đó, [7]-[9] đưa ra thiết kế sử dụng các khối FFT 8 hoặc 16 điểm cùng các khối nhân, thanh ghi lưu trữ để đạt được thiết kế 64/128/256 điểm với tốc độ tính toán nhanh hơn. Do đó, việc thiết kế và tối ưu FFT 16 điểm có ý nghĩa thực tiễn trong việc thực thi FFT cho các ứng dụng hiện nay. Ngoài ra, thiết kế trong bài báo [10] chỉ thực hiện cho việc tính toán FFT với số nguyên và có sai số cao hoặc thiết kế trong các bài báo [11] và [12] tồn tại nhược điểm là tần số hoạt động khá thấp. Do đó, trong bài báo này, chúng tôi đề xuất thiết kế bộ tính toán FFT với số phức và có sai số ít hơn so với thiết kế trong bài báo [10]. Thiết kế đề xuất cũng đòi hỏi tài nguyên logic ít hơn 25% so với thiết kế trong bài báo [10] và [11], trong khi có tần số hoạt động tối đa cao hơn 2.4 lần so với thiết kế trong [10], 6 lần so với thiết kế trong [11] và 2.8 lần so với thiết kế trong [12].

## 2. Cơ sở lý thuyết

### 2.1. Biến đổi Fourier nhanh cơ số 2

DFT  $N$  điểm thực hiện với  $N^2$  phép nhân và  $N(N - 1)$  phép cộng dẫn đến việc tính toán trực tiếp từ DFT trên phần cứng có những bất lợi khi khối lượng phép tính lớn, cần nhiều tài nguyên và tốc độ thấp. FFT [1] đã được triển khai chỉ với  $\frac{N}{2} \log_2 N$  phép nhân và  $N \log_2 N$  phép cộng tức đã giảm đi rất nhiều so với DFT. Không dừng lại ở đó, lần lượt từng thuật toán FFT khác nhau được ra đời với mong muốn tối ưu về tài nguyên và tốc độ như cơ số 2 phân chia theo thời gian hay tần số, cơ số 4, trộn cơ số ... Trong nội dung bài báo này, chúng tôi triển khai theo cơ số 2 phân chia theo tần số [1].

### 2.2. Các kiến trúc pipeline FFT

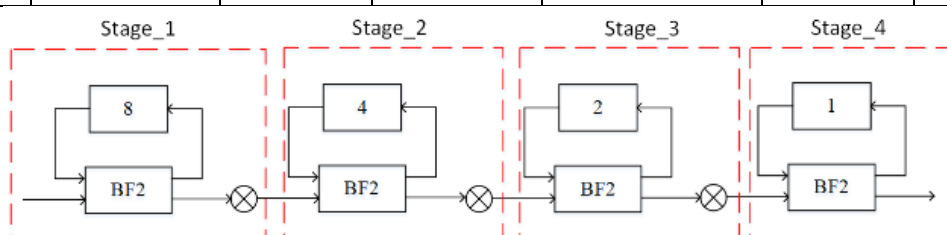
Pipeline FFT là một nhóm các kiến trúc thuật toán FFT đặc biệt để có thể tính toán DFT một cách tuần tự. Nó thỏa mãn yêu cầu về thời gian thực và xử lý liên tiếp khi dữ liệu được cung cấp liên tục thông qua bộ xử lý. Một số kiến trúc khác nhau đã được đề xuất dựa trên loại thuật toán theo cơ số và sự phân chia. Các đặc điểm chính và tài nguyên của một số kiến trúc pipeline FFT được trình bày ở Bảng 1. Có thể thấy rằng 2 kiểu kiến trúc R2SDF (Radix-2 Single-path Delay Feedback) và R2MDC (Radix-2 Multipath Delay Commutation) dựa trên thuật toán cơ số 2 có độ phức tạp khi thiết kế là nhỏ nhất nên dễ tiếp cận nhất. Chúng có cùng thông số tối ưu hoạt động là 50%, tức là với 16 chu kỳ xung clock thì bộ cộng hoặc bộ nhân hoạt động với 8 chu kỳ. Tuy nhiên R2SDF yêu cầu sử dụng ít thanh ghi hơn nên sẽ tiết kiệm về tài nguyên phần cứng hơn. Do đó, chúng tôi chọn kiến trúc R2SDF để thiết kế bộ tính toán FFT 16 điểm.

### 2.3. Kiến trúc R2SDF

R2SDF là một kiến trúc pipeline FFT đơn đường hồi tiếp theo thuật toán cơ số 2. Dữ liệu sẽ được đưa vào tuần tự từng điểm và được tính toán liên tục. Cấu trúc luồng dữ liệu R2SDF FFT 16 điểm được minh họa như trong hình 1 và chia ra 4 tầng. Các khối BF2 là bướm 2 điểm, các khối 8, 4, 2, 1 là các thanh ghi FIFO (First-In First-Out) và cuối cùng là các khối nhân với thừa số xoay. Dữ liệu sẽ được luân chuyển như sơ đồ bướm, bắt đầu là 8 điểm  $x(n)$  đưa vào FIFO tầng đầu tiên, sau đó 8 điểm  $x(n)$  tiếp theo được tính toán bướm với 8 điểm đầu, kết quả phép cộng được đưa thẳng qua tầng tiếp theo còn phép trừ thì lưu vào FIFO. Khi trong FIFO có đủ 8 kết quả phép trừ thì sẽ được đưa qua bộ nhân và đến tầng tiếp theo. Các tầng khác cũng thực hiện tương tự.

**Bảng 1.** So sánh tài nguyên phần cứng của kiến trúc pipeline FFT [13]

Kiến trúc	Bộ nhân phức	Bộ cộng phức	Thanh ghi	Độ phức tạp khối điều khiển	Tối ưu hoạt động	
					Bộ cộng	Bộ nhân
R2SDF	$\log_2 N - 2$	$2\log_2 N$	$N - 1$	Đơn giản	50%	50%
R2MDC	$\log_2 N - 2$	$2\log_2 N$	$3N/2 - 2$	Đơn giản	50%	50%
R2 <sup>2</sup> SDF	$\log_4 N - 1$	$4\log_4 N$	$N - 1$	Trung bình	75%	75%
R4SDF	$\log_4 N - 1$	$8\log_4 N$	$N - 1$	Trung bình	25%	75%
R4MDC	$3(\log_4 N - 1)$	$8\log_4 N$	$5N/2 - 4$	Trung bình	25%	25%
R4SDC	$\log_4 N - 1$	$3\log_4 N$	$2N - 2$	Phức tạp	100%	75%

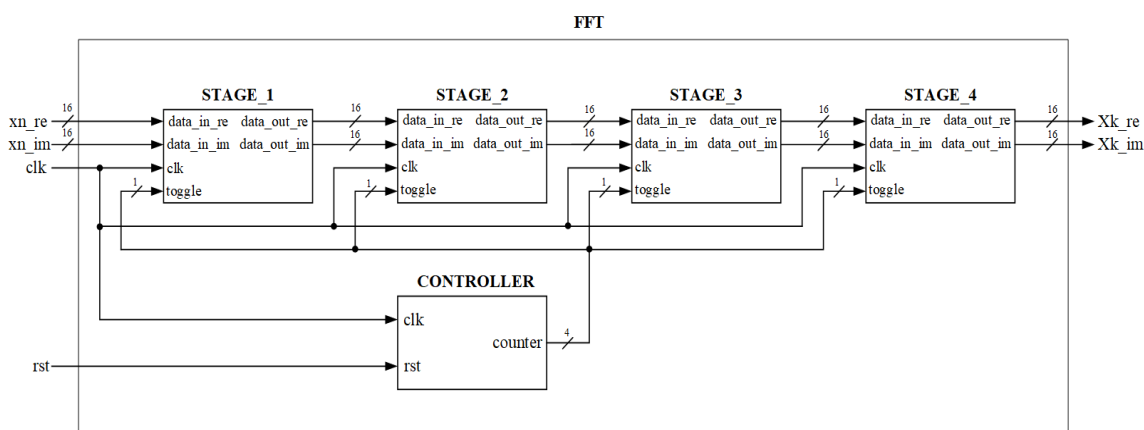


**Hình 1.** Kiến trúc R2SDF FFT 16 điểm.

### 3. Thiết kế hệ thống

#### 3.1. Sơ đồ khối toàn hệ thống

Từ kiến trúc R2SDF, hệ thống được thiết kế tương tự với 4 khối Stage và 1 khối Controller để điều khiển, được thể hiện như hình 2. Đường dữ liệu chính của hệ thống là phần thực và phần ảo của các điểm FFT với mỗi phần sẽ được tính toán với kích thước 16 bit, kết nối xuyên suốt qua các tầng. Các khối trong hệ thống đều sử dụng xung clock đồng bộ và các tầng sẽ chịu sự điều khiển của tín hiệu toggle cùng lúc. Ngoài ra còn có tín hiệu rst (reset) bất đồng bộ để buộc bộ tính toán thực thi lại từ đầu với dữ liệu mới đưa đến ở ngõ vào.

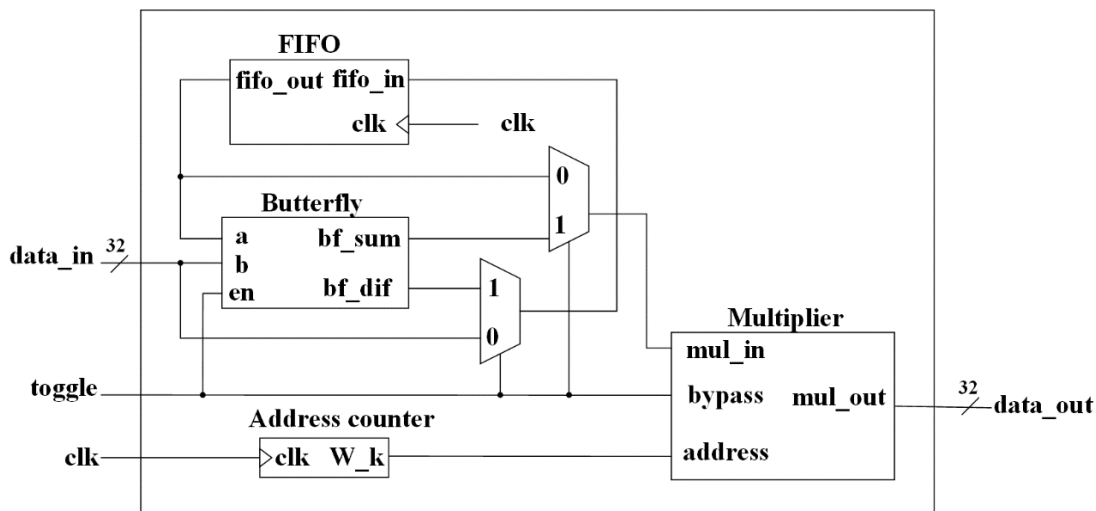


**Hình 2.** Sơ đồ khối hệ thống FFT được thiết kế

#### 3.2. Khối Stage

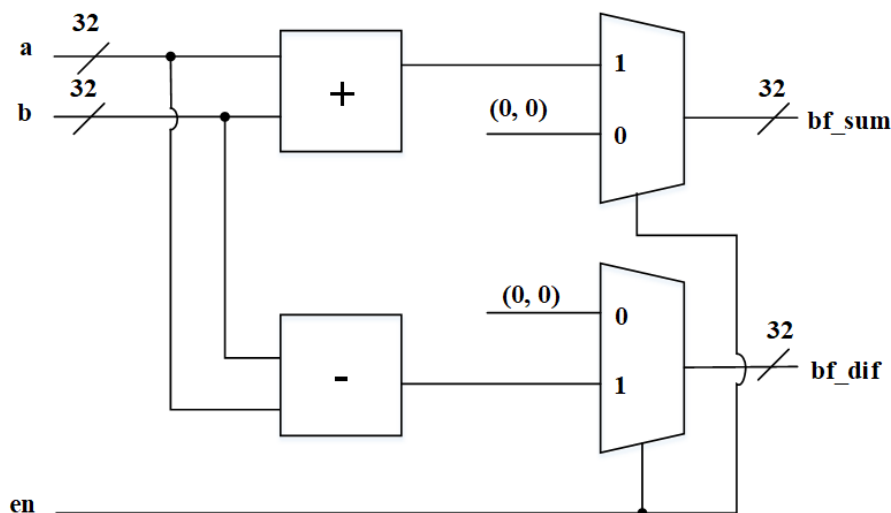
Khối này gồm module Butterfly (bướm), FIFO, Multiplier (nhân với thừa số xoay) và ROM (lưu các giá trị thừa số xoay) được minh họa như hình 3. Khối Stage có 2 trạng thái hoạt động được lựa chọn bởi tín hiệu toggle. Khi toggle = 0, dữ liệu ngõ vào sẽ được đưa vào FIFO và ngõ ra FIFO sẽ đưa đến bộ

nhân với thừa số xoay tương ứng, đây là trạng thái một nửa dữ liệu đầu tiên. Khi toggle = 1, dữ liệu ngõ vào sẽ đưa đến bướm tính toán, kết quả phép cộng sẽ được đưa thẳng ra data\_out, kết quả phép trừ sẽ được lưu vào FIFO và chờ cho đến trạng thái toggle = 0 tiếp theo để được nhân với thừa số xoay.



Hình 3. Sơ đồ khối của khối Stage

### 3.3. Module Butterfly



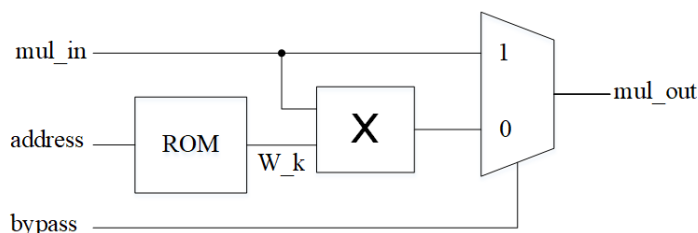
Hình 4: Sơ đồ khối của module Butterfly.

Module Butterfly có nhiệm vụ thực hiện phép cộng trừ trong sơ đồ bướm 2 điểm. Tín hiệu ngõ vào là a, b và ngõ ra là bf\_sum, bf\_dif, mỗi tín hiệu đều bao gồm phần thực và phần ảo. Tuy nhiên trong quá trình thực thi thì không phải lúc nào bướm cũng cần thực hiện phép tính. Vậy nên module cần có một tín hiệu en cho phép thực thi, nếu en=0 thì kết quả ngõ ra sẽ bằng 0. Module Butterfly bao gồm một bộ cộng, một bộ trừ và hai bộ đa hợp được điều khiển bởi tín hiệu cho phép, có sơ đồ RTL được thể hiện như hình 4.

### 3.4. Module FIFO

FIFO là thanh ghi dịch, với thứ tự của dữ liệu là vào trước ra trước. Ở trong hệ thống này, FIFO không cố định kích thước mà thay đổi dựa theo vị trí tầng. Ở tầng 1, kích thước của FIFO sẽ là 8 ứng với  $2^3$  số phức, mỗi số 32 bit (16 bit cho phần thực và 16 bit cho phần ảo), tầng 2 sẽ là  $2^2 = 4$ , tầng 3 là 2 và tầng 4 là 1 số phức. Như vậy kích thước FIFO của mỗi tầng sẽ bằng  $2^{4-i}$  với i là số tầng.

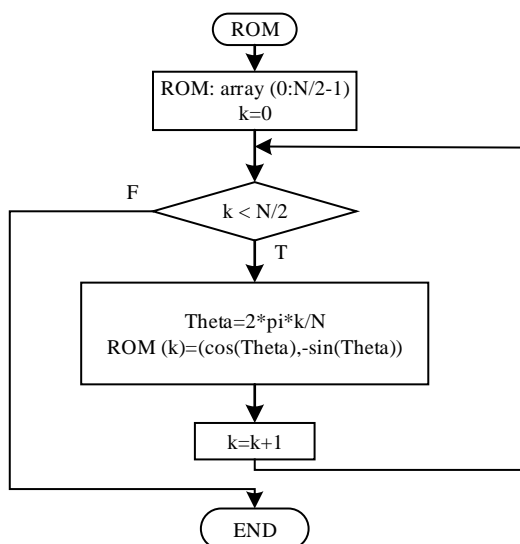
### 3.5. Module Multiplier



**Hình 5.** Sơ đồ module Multiplier

Khối Multiplier có sơ đồ RTL như hình 5 thực hiện phép nhân dữ liệu với thừa số xoay. Vì không phải lúc nào dữ liệu cũng cần nhân nên cần có một tín hiệu điều khiển cho phép bỏ qua bypass. Thừa số xoay sẽ được lưu trữ trong module ROM (sẽ trình bày ở phần kế tiếp) và chọn ra bởi tín hiệu address. Lúc này dữ liệu đầu vào *mul\_in* sẽ được nhân với *W<sub>k</sub>* lấy ra từ module ROM bằng bộ nhân số phức. Việc lựa chọn địa chỉ cho module ROM sẽ được trình bày trong phần module Address counter.

### 3.6. Module ROM



**Hình 6:** Lưu đồ module ROM.

Module ROM là bộ nhớ lưu trữ giá trị của các thừa số xoay. Vì FFT *N* điểm đã giảm còn *N/2* thừa số so với DFT nên ta chỉ cần tính toán và lưu trữ các giá trị  $W_N^k$  với  $k = 0, 1, \dots, \frac{N}{2} - 1$ . Lưu đồ thuật toán ROM được thể hiện ở hình 6. Trong đó ROM được khởi tạo với kiểu mảng các số phức. Sau đó, một vòng lặp được thực thi để tính giá trị *Theta* là lũy thừa của  $e^{-j\frac{2\pi k}{N}}$  và được biến đổi sang công thức Euler  $e^{-j\theta} = \cos(\theta) - j\sin(\theta)$  để lưu vào ROM.

### 3.7. Module Address counter

Trong thiết kế khối Stage, ở mỗi cạnh lên xung clock, dữ liệu ngõ ra sẽ thay đổi phụ thuộc vào dữ liệu ngõ vào và dữ liệu trong FIFO. Do trạng thái của khối Stage thay đổi mỗi *N/2* chu kỳ xung clock và địa chỉ cần dùng từ 0 đến *N/2 - 1* nên Address counter là bộ đếm mod *N/2*. Giá trị khởi tạo của biến đếm là 0, counter sẽ tăng mỗi khi xuất hiện cạnh lên xung clock và trở về 0 khi đếm đến *N/2*.

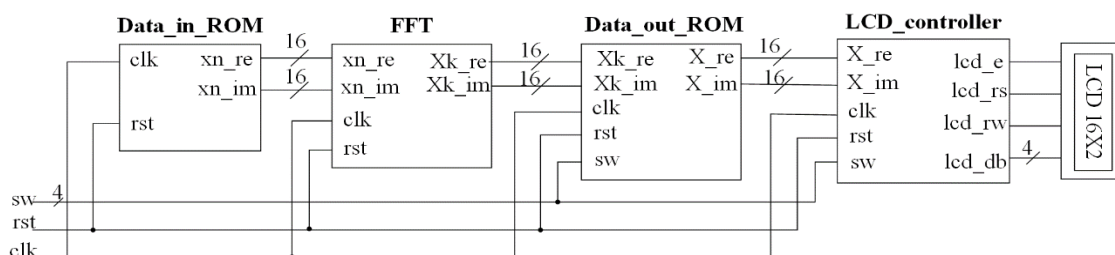
### 3.8. Khối Controller

Qua thiết kế khối Stage ở trên có thể thấy rằng Stage<sub>1</sub> cần đổi trạng thái toggle sau 8 chu kỳ xung clock, tương ứng với các khối Stage còn lại là 4, 2, 1 xung clock. Như vậy khối Controller là một mạch

đếm 4 bit với bit trọng số cao sẽ điều khiển khối Stage đầu tiên, bit trọng số thấp sẽ điều khiển khối Stage cuối cùng.

### 3.9. Mạch kiểm tra

Sơ đồ khối mạch kiểm tra được thể hiện ở hình 7. Các khối Data\_in\_ROM và Data\_out\_ROM có nhiệm vụ lưu trữ và đẩy dữ liệu  $x(n)$  ở ngõ vào và lưu kết quả ở ngõ ra rồi sắp xếp lại theo thứ tự các  $X(k)$ . Khối LCD\_Controller có chức năng hiển thị giá trị  $X(k)$  được lựa chọn bằng 4 switch được đếm theo nhị phân.



Hình 7. Sơ đồ khối mạch kiểm tra

## 4. Kết quả đánh giá qua mô phỏng và trên kit FPGA thực tế

Bài báo sử dụng dữ liệu ngõ vào như sau để đánh giá kết quả:  $xn=\{(1,-1),(2,-2),(3,-3),\dots,(16,-16)\}$

### 4.1. Tài nguyên sử dụng

Bảng 2. Tài nguyên bộ tính FFT 16 điểm

Logic Utilization	Used	Available	Utilization
Number of Slices	1803	4656	38%
Number of Slices Flip Flops	156	9312	1%
Number of 4 input LUTs	3127	9312	33%
Number of bonded IOBs	66	232	28%
Number of MULT18X18SIOs	12	20	60%
Number of GCLKs	1	24	4%

Bảng 3. Tần số hoạt động tối đa bộ tính toán FFT 16 điểm

Speed Grade	-4
Maximum Frequency	149.867 MHz
Minimum input arrival time before clock	7.452 ns
Maximum output required time after clock	51.977 ns
Maximum combinational path delay	52.564 ns

Bảng 2 tóm tắt tài nguyên cần sử dụng cho bộ tính toán FFT 16 theo kiến trúc pipeline R2SDF khi được tổng hợp dùng phần mềm Xilinx ISE 14.7, giả sử chọn dòng chip FPGA XC3S500E. Tần số xung clock tối đa có thể hoạt động là 149.867 MHz. Như vậy trong 1 giây, bộ tính toán này có thể thực hiện  $(149.867 \cdot 10^6) / 32 = 4,683,343$  phép tính FFT 16 điểm.

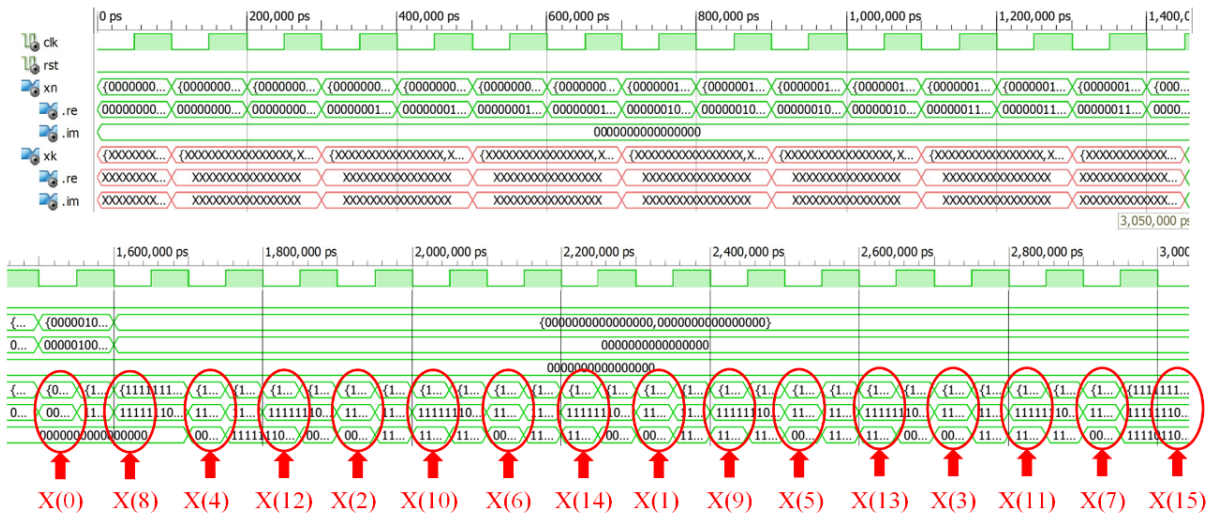
Bảng 4. So sánh với các thiết kế khác

	Thiết kế đề xuất	Abhishek Mankar [10]	Josue Saenz [11]	Parvin and Hussain [12]
Slices	1803	2389	2402	x
Flip Flops	156	1913	295	x
4 input LUTs	3127	3972	x	223
IOBs	66	x	51	x
Fmax (MHz)	149.867	61.831	25	54.089

Kết quả so sánh thiết kế được đề xuất trong bài báo này với các thiết kế đã có được thể hiện trong bảng 4. Có thể thấy rằng tài nguyên của thiết kế đề xuất so với thiết kế ở [10] [11] giảm khoảng 25%, đồng thời tần số hoạt động tối đa (Fmax) cũng lớn hơn 2.4 lần so với [10] và 6 lần so với [11]. So với thiết kế [12], mặc dù thiết kế đề xuất sử dụng nhiều tài nguyên hơn nhưng tạo ra tần số hoạt động lớn hơn 2.8 lần thiết kế cũ.

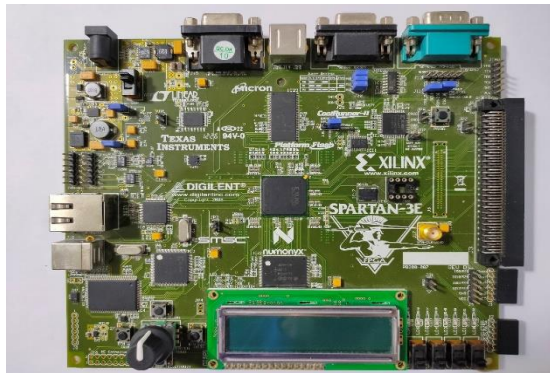
**4.2. Kết quả đánh giá chức năng qua mô phỏng**

Sau khi chạy mô phỏng với dữ liệu ngõ vào đã trình bày, dạng sóng thu được minh họa trong hình 8.



**Hình 8.** Dạng sóng tín hiệu tính toán FFT 16 điểm.

**4.3. Kết quả đánh giá thực tế trên kit FPGA**



**Hình 9.** Phần cứng thực tế



**Hình 10.** Kết quả thực tế

Mạch kiểm tra được tổng hợp và nạp xuống kit FPGA Spartan-3E như hình 9, kết quả được thể hiện trên hình 10. Kết quả thực thi giữa mô phỏng và thực tế không cho thấy sự chênh lệch nào. Để so sánh kết quả tổng quát hơn, kết quả ngõ ra  $X(k)$  của hệ thống được liệt kê và so sánh với kết quả lý thuyết được tính bằng MATLAB ở bảng 5 với sai số ghi nhận được tối đa là 0.2853.

## 5. Kết luận

Qua quá trình đánh giá kết quả thực thi của bộ tính toán FFT 16 điểm, có thể thấy rằng thiết kế hoạt động tốt, kết quả cho ra tương đối chính xác so với lý thuyết từ mô phỏng cho đến đánh giá thực tế trên kit FPGA. Tài nguyên hệ thống sử dụng ít hơn khi so sánh với 2 thiết kế được xem xét từ đầu là [10] và [11] khoảng 25%, cũng như đạt được tần số hoạt động khá cao là 149.867 MHz cho ra tốc độ biến đổi Fourier nhanh 16 điểm là 4,683,343 lần mỗi giây. Thiết kế được trình bày đã khắc phục nhược điểm tần số thấp và chỉ dùng số nguyên dẫn đến sai số nhiều của các nghiên cứu trước đó. Ngoài ra, việc thiết kế theo kiến trúc pipeline có thể dễ dàng mở rộng hoặc sử dụng trong các bộ tính toán FFT nhiều điểm hơn.

Các ứng dụng hiện nay không chỉ sử dụng mỗi FFT mà còn IFFT (Inverse Fast Fourier Transform) nên chúng ta có thể tích hợp thêm bộ IFFT với những module có sẵn và có thể linh hoạt chuyển đổi với 1 tín hiệu điều khiển ngõ vào. Ngoài ra, việc cải tiến và nâng cấp là điều bắt buộc khi dùng các board mạch FPGA được tích hợp các khối xử lý DSP nhằm làm tăng tốc độ xử lý, giảm bớt lượng tài nguyên cần thiết, hỗ trợ dấu chấm động cho ra kết quả chính xác hơn. Không những thế, việc nghiên cứu và ứng dụng các thuật toán FFT phức tạp hơn như cơ số 4, 8 hay thậm chí 16 sẽ giúp các bộ tính toán FFT nhiều điểm hoạt động hiệu quả hơn.

**Bảng 5.** So sánh kết quả thực tế và lý thuyết bộ tính toán FFT 16 điểm được thiết kế.

$X(k)$	Thực tế	Lý thuyết	Chênh lệch $ \Delta X_k $
X(0)	(136.0, -136.0)	(136.0, -136.0)	(0.0, 0.0)
X(1)	(32.0, 48.0)	(32.2187, 48.2187)	(0.2187, 0.2187)
X(2)	(11.25, 27.25)	(11.3137, 27.3137)	(0.0637, 0.0637)
X(3)	(3.6875, 19.6875)	(3.9728, 19.9728)	(0.2853, 0.2853)
X(4)	(0.0, 16.0)	(0.0, 16.0)	(0.0, 0.0)
X(5)	(-2.5, 13.5)	(-2.6546, 13.3454)	(0.1546, 0.1546)
X(6)	(-4.75, 11.25)	(-4.6863, 11.3137)	(0.0637, 0.0637)
X(7)	(-6.5, 9.5)	(-6.4087, 9.5913)	(0.0913, 0.0913)
X(8)	(-8.0, 8.0)	(-8.0, 8.0)	(0.0, 0.0)
X(9)	(-9.5, 6.5)	(-9.5913, 6.4087)	(0.0913, 0.0913)
X(10)	(-11.25, 4.75)	(-11.3137, 4.6863)	(0.0637, 0.0637)
X(11)	(-13.1875, 2.8125)	(-13.3454, 2.6546)	(0.1579, 0.1579)
X(12)	(-16.0, 0.0)	(-16.0, 0.0)	(0.0, 0.0)
X(13)	(-20.0, -4.0)	(-19.9728, -3.9728)	(0.0272, 0.0272)
X(14)	(-27.25, -11.25)	(-27.3137, -11.3137)	(0.0637, 0.0637)
X(15)	(-48.0, -32.0)	(-48.2187, -32.2187)	(0.2187, 0.2187)

## TÀI LIỆU THAM KHẢO

- [1] J. Cooley and J. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, vol. 19, pp. 297-301, Apr. 1965.
- [2] C. Lin, Y. Yu and L. Van, "A Low-Power 64-Point FFT/IFFT Design for IEEE 802.11a WLAN Application", IEEE International Symposium on Circuits and Systems, pp. 4523-4526, May. 2006.
- [3] P. Wang, J. McAllister and Y. Wu, "Software Defined FFT Architecture for IEEE 802.11ac", 2013 IEEE Global Conference on Signal and Information Processing, pp. 1246-1249, Dec. 2013.



- [4] P. Dinh, L. Lanante, M. Nguyen, M. Kurosaki and H. Ochi, "An Area-Efficient Multimode FFT Circuit for IEEE 802.11ax WLAN Devices", 2017 19th International Conference on Advanced Communication Technology, pp. 735-739, Feb. 2017.
- [5] J. Zyren, "Overview of the 3GPP Long Term Evolution Physical Layer", Freescale Semiconductor, Jul. 2007
- [6] B. Zhou, Y. Peng and D. Hwang, "Pipeline FFT Architectures Optimized for FPGAs", International Journal of Reconfigurable Computing, vol. 2009, Sep. 2009.
- [7] K. Maharatna, E. Grass, and U. Jagdhold, "A 64-Point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM", IEEE Journal of Solid-State Circuits, vol. 39, no. 3, pp. 484-493, Mar. 2004.
- [8] B. Fernandes and H. Sarmiento, "FPGA Implementation and Testing of A 128 FFT for A MB-OFDM Receiver", Analog Integrated Circuits and Signal Processing, vol. 70, pp. 241-248, Sep. 2011.
- [9] S. Gupta, "Low Power Implementation of Fast Fourier Transform Processor on FPGA", International Journal of Advanced Computer Research, vol. 3, no. 4, pp. 98-105, Dec. 2013.
- [10] Abhishek Mankar, A. Diptisankar Das and N. Prasad, "FPGA Implementation of 16-Point Radix-4 Complex FFT Core Using NEDA", Students Conference on Engineering and Systems (SCES), pp. 1-5, 2013.
- [11] S. Saenz, J. Raygoza, E. Becerra, S. Cisneros and J. Dominguez, "FPGA Design and Implementation of Radix-2 Fast Fourier Transform Algorithm with 16 and 32 Points", 2015 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC), pp. 1-6, 2015.
- [12] K. N. Parvin and M. Z. Hussain, "Impact of radices for the design of efficient FFT processor," 2018 2nd International Conference on Inventive Systems and Control (ICISC), pp. 950-954, 2018
- [13] S. He and M. Torkelson, "A new approach to pipeline FFT processor", Proceedings of the 10th International Parallel Processing Symposium, pp. 766-770, Apr. 1996.
- [14] M. Garrido, M. Acevedo, A. Ehliar and O. Gustafsson, "Challenging the Limits of FFT Performance on FPGAs", 2014 International Symposium on Integrated Circuits (ISIC), pp. 172-175, 2014.
- [15] V. Patil and T. M. Manu, "FPGA Implementation Radix-2 DIT FFT Using Fixed Point Arithmetic and Reduced Arithmetic Complexity," 2021 International Conference on Intelligent Technologies (CONIT), 2021.
- [16] J. Wang, Y. Xie, B. Li, C. Yang and S. Hu, "The Reconfigurable Pipelined Variable-point FFT Processor Design," 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019.
- [17] S. Sanjeet, B. D. Sahoo and K. K. Parhi, "Comparison of Real-Valued FFT Architectures for Low-Throughput Applications using FPGA," 2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS), 2021.



**Nguyen Van Thanh Loc** received his B.S degree from Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam, in 2020. His main research interests include communication networks and applications of error-control coding for wireless communications.



**Hoang Trung Huu Thuong** received his B.S degree from Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam, in 2021. His main research interests include communication networks and FPGA-based design for DSP applications.



**Mai Thi Hoai Nhi** received his B.S degree from Ho Chi Minh City University of Technology and Education (HCMUTE), Vietnam, in 2021. Her main research interests include wireless communication networks and FPGA-based designs for DSP applications.



**Do Duy Tan** received his B.S. degree from Ho Chi Minh City University of Technology (HCMUT), Vietnam, and M.S. degree from Kumoh National Institute of Technology, Korea, in 2010 and 2013, respectively. He received his Ph.D. degree from Autonomous University of Barcelona, Spain, in 2019. He is currently with the Department of Computer and Communication Engineering, Ho Chi Minh City University of Technology and Education (HCMUTE) in Vietnam as an Assistant Professor. His main research interests include real-time optimisation for resource allocation in wireless networks and coding applications for wireless communications.