

THIẾT KẾ PID TỐC ĐỘ CAO PIPELINE ĐA TẦNG KIỂM SOÁT TỐC ĐỘ ĐỘNG CƠ DC TRÊN FPGA

Bùi An Đông, Huỳnh Quốc Thịnh, Bùi Trọng Tú

Trường Đại học Khoa học Tự nhiên, ĐHQG-HCM

hqthinh@fetel.hcmus.edu.vn, badong@hcmus.edu.vn, btuu@hcmus.edu.vn

TÓM TẮT: Bài báo này trình bày kiến trúc VLSI của PID (Proportional-Integral-Derivative) tốc độ cao pipeline đa tầng để điều khiển tốc độ động cơ DC theo thời gian thực trên FPGA. Nhóm tác giả thiết kế PID theo hướng thiết kế số, với dữ liệu đầu vào 16 bit bao gồm giá trị các hệ số K_p , K_i , K_d và giá trị ngõ vào, kết hợp mạch nhân nhanh Booth multiplier, bộ cộng Carry Look Ahead và xử lý pipelined, cho kết quả ngõ ra 32 bit (có dấu). Thiết kế PID này được đánh giá bằng việc tổng hợp trên FPGA Cyclone II với tần số 148 MHz và xây dựng mô hình kiểm soát ổn định vận tốc quay động cơ DC trên kit FPGA đáp ứng thời gian thực đạt tần số 128MHz.

Từ khóa: PID, Booth multiplier, CLA, Pipelined.

I. GIỚI THIỆU

Thuật toán điều khiển PID (Proportional-Integral-Derivative) đã được ứng dụng hơn 70 năm và là một trong những phương pháp thiết kế hệ thống điều khiển của lịch sử lâu nhất [1]. PID có những ưu điểm do có cấu trúc đơn giản, độ ổn định tốt và hiệu quả. Công nghệ FPGA (Field Programmable Gate Array) ngày càng được sử dụng rộng rãi trong thiết kế số nhờ có tính linh động, khả năng tái cấu hình và tốc độ ngày càng cao, có khả năng đáp ứng thời gian thực [2]. FPGA đã được sử dụng trong các hệ thống điều khiển chuyển động robot, trong vòng điều khiển PID [3]. Thiết kế trong nghiên cứu [1] sử dụng ADC 12 bit để chuyển đổi vận tốc động cơ và DAC 8 bit để giao tiếp điều khiển ở ngõ ra và cần đến 2 bộ nhân, 3 bộ cộng để hoàn thành việc tính PID. Trong khi đó, nghiên cứu [3] chỉ có ngõ vào 12 bit, cần 3 bộ nhân 12 bit và 5 bộ cộng 24 bit để cho kết quả ngõ ra 26 bit.

Trong bài báo này, chúng tôi sẽ trình bày thiết kế bộ điều khiển PID tốc độ cao pipeline đa tầng theo định hướng VLSI, trong đó sử dụng 1 bộ nhân nhanh Booth radix-4 và 1 bộ cộng pipeline 2 tầng giúp tiết kiệm tài nguyên nhưng vẫn đạt tốc độ cao. Thiết kế PID sau đó được thực hiện trên FPGA nhằm kiểm soát tốc độ động cơ DC theo thời gian thực.

Nội dung bài báo gồm các phần sau đây. Thuật toán và mô hình số hóa PID được trình bày trong phần II. Phần III trình bày mô hình thiết kế PID và chu trình hoạt động; thực nghiệm mô hình điều khiển vận tốc động cơ DC theo thời gian thực được trình bày trong phần IV. Kết luận được đưa ra trong phần V.

II. THUẬT TOÁN PID

PID là một bộ điều khiển hồi tiếp vì tích phân tỉ lệ được sử dụng rộng rãi trong các hệ thống điều khiển công nghiệp. Thuật toán PID tính dựa vào sự sai khác giữa tín hiệu mong muốn $x(t)$ (Set Point -SP) và tín hiệu đã được tính toán $y(t)$ (Processed Value - PV) để điều chỉnh hệ thống sao cho lỗi tín hiệu là nhỏ nhất. Thuật toán điều khiển PID bao gồm 3 thành phần cơ bản là: Tính toán tỉ lệ Proportional (P) $K_p e(t)$, tính toán tích phân Integral (I) $K_i \int_0^t e(t) dt$ và tính toán vi phân Derivative (D) $K_d \frac{de(t)}{dt}$.

Mục tiêu của PID là tạo ra $u(t)$ để điều khiển hệ thống sao cho tín hiệu $y(t)$ càng giống tín hiệu mong muốn $x(t)$ càng tốt. Đầu vào của PID là sự sai khác giữa tín hiệu lý tưởng $x(t)$ và ngõ ra thực tế $y(t)$. Biểu thức (1) thể hiện ngõ ra của khối PID trong miền thời gian, trong đó K_p , K_i , K_d lần lượt là các hệ số tỉ lệ, hệ số tích phân và hệ số vi phân. Các hệ số K_p , K_i , K_d được lựa chọn sao cho hệ thống có thể đáp ứng nhanh nhất.

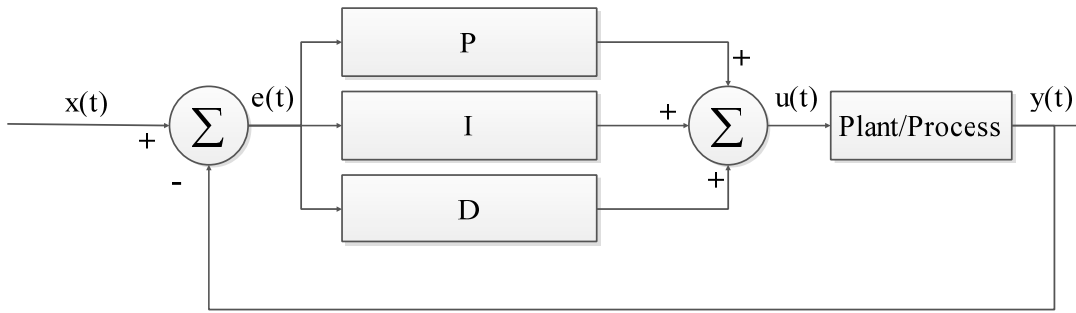
$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (1)$$

Hệ thống PID thông thường, biểu thức (1), được minh họa trong Hình 1, với:

$$P = K_p e(t)$$

$$I = K_i \int_0^t e(t) dt$$

$$D = K_d \frac{de(t)}{dt}$$

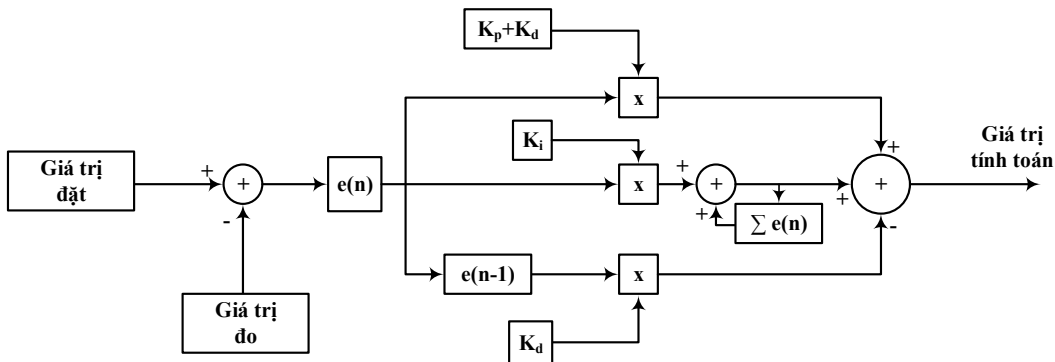


Hình 1. Mô hình hệ thống PID

Trong các ứng dụng số, ta thay dt trong biểu thức (1) bằng chu kỳ lấy mẫu có giá trị rất nhỏ, để rời rạc hóa biểu thức (1), ta được hệ thống PID ở biểu thức (2) có giá trị gần bằng giá trị trong biểu thức (1).

$$u(n) = (K_p + K_d)e(n) + K_i \sum_{j=1}^n e(j) - K_d e(n - 1) \tag{2}$$

Từ biểu thức (2) ta xây dựng nên sơ đồ hệ thống cơ bản như Hình (2). Hệ thống bao gồm 3 bộ nhân cho các phép tính $(K_p+K_d)*e(n)$, $K_i*e(n) + K_i*e(n-1)$ và $K_d*e(n-1)$, 3 bộ cộng cho phép tính lỗi giữa $x(t)$ và $y(t)$, tính $K_i \sum_{j=1}^n e(j)$ và tính tổng tín hiệu ngõ ra $u(t)$.



Hình 2. Mô hình rời rạc hóa PID

Thuật toán PID trên phần cứng được minh họa bởi đoạn mã giả sau:

```

While(1){
    error =Input - Setpoint;
    errSum +=error;
    output = (kp+kd) *error +ki *errSum - kd *lastErr;
    lastErr =error;
    delay_ms(100);
}
    
```

Với :

error là e(n),

Input là giá trị đo,

Setpoint là giá trị cân bằng mong muốn,

Chu kỳ tính toán là 100ms.

III. THIẾT KẾ PID

Mô hình đề xuất được thể hiện ở Hình 3, bao gồm 1 bộ nhân và 1 bộ cộng tính toán pipeline 2 tầng, 2 mảng thanh ghi, 4 bộ mux và bộ điều khiển hoạt động hệ thống.

Khối thanh ghi

Bao gồm các thanh ghi 16 bit chứa các hệ số KP, KI, KD, KPD, ERR0, ERR1 và khối thanh ghi 64 bit là UN, SIGMA. Chi tiết các thanh ghi được chỉ ra ở Bảng 1.

Bộ mux

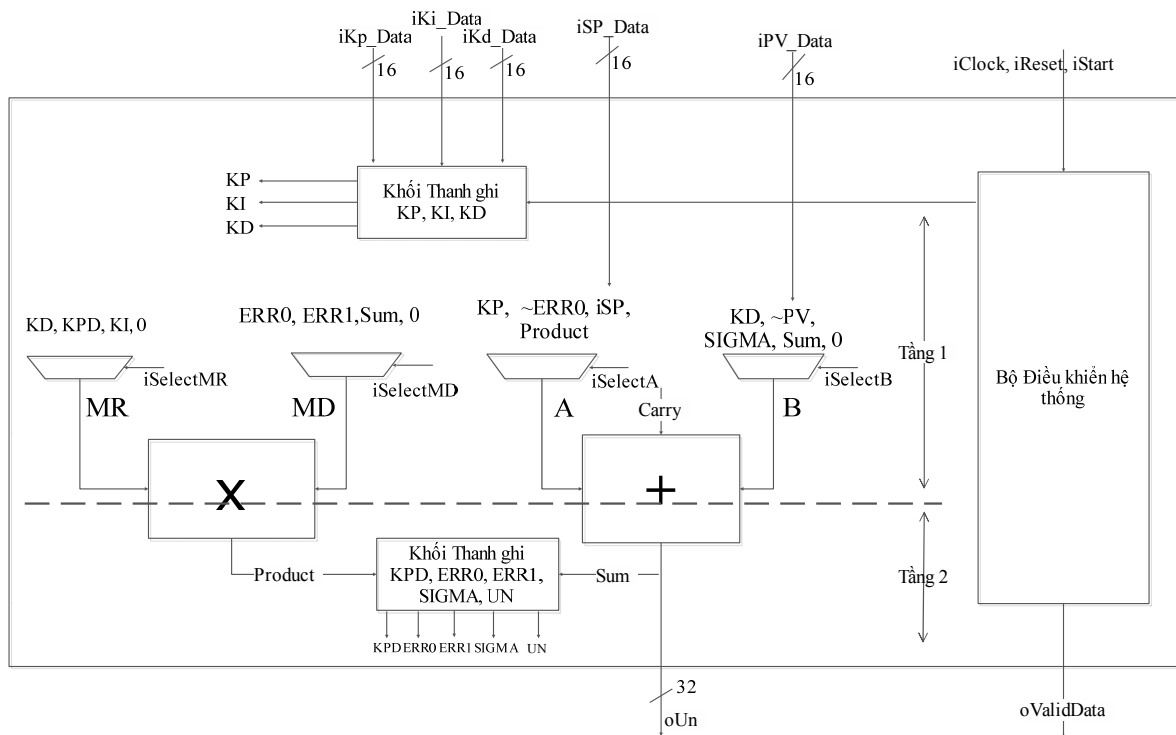
Có 4 bộ mux để điều khiển lối vào cho bộ cộng và bộ nhân. Bộ mux MR có 4 ngõ vào KD, KPD, KI và 0; bộ mux MD có 4 ngõ vào ERR0, ERR1, Sum, 0; bộ mux A có 4 ngõ vào KP, ~ERR0, iSP, Product; bộ mux B có 5 ngõ vào KD, ~PV, SIGMA, Sum, 0. Các tín hiệu chọn lựa kết quả ngõ ra của các bộ mux được *bộ điều khiển hệ thống* điều khiển tương ứng với phép toán thực hiện trong bảng 2.

Bộ cộng và nhân

Bộ cộng được sử dụng là bộ cộng Carry Look Ahead 32 bit cho tốc độ cao và tiết kiệm diện tích. Bộ nhân 16x16 được xây dựng trên thuật toán Booth Encoder Array [4]. Cả bộ cộng và bộ nhân được thiết kế 2 tầng bao gồm tầng thanh ghi và tầng tính toán. Mỗi bộ lần lượt thực hiện các phép toán theo xử lý pipeline, để tăng tốc độ tính toán như trình bày trong bảng 2. Tầng 1, tầng thanh ghi lưu trữ giá trị chuẩn bị cho tính toán; tầng 2, tầng thực hiện phép toán cộng hoặc nhân.

Bảng 1. Chức năng của các thanh ghi

Tên	Độ rộng bit	Mô tả
KP	16	Lưu hệ số Kp
KI	16	Lưu hệ số Ki
KD	16	Lưu hệ số Kd
KPD	16	Giá trị Kp+Kd
ERR0	16	Lưu tín hiệu lỗi e(n)
ERR1	16	Lưu tín hiệu lỗi e(n-1)
UN	32	Lưu tín hiệu u(n)
SIGMA	32	Lưu giá trị $K_i \sum_{j=1}^n e(j)$

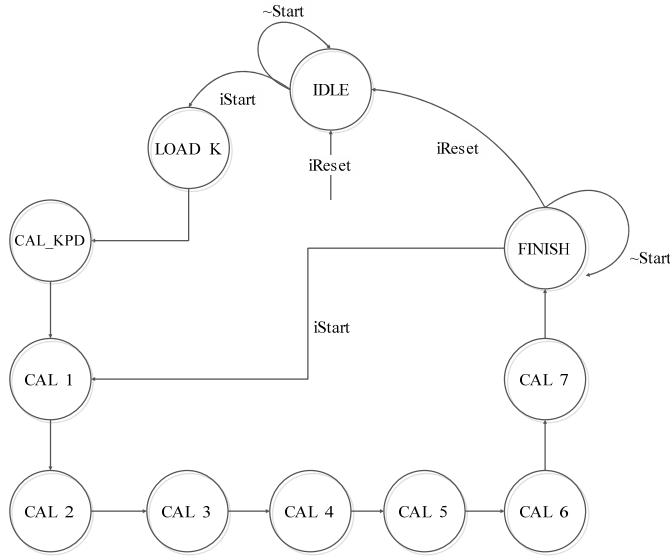


Hình 3. Mô hình kiến trúc VLSI của thiết kế PID

Hoạt động

Bởi vì chỉ sử dụng một bộ cộng và một bộ nhân cho các tất các các phép tính nên ta phải phân chia thời gian và nhiệm vụ cho hai bộ này. Bảng 2 mô tả hoạt động của tính toán u(n) của bộ cộng và bộ nhân.

Các hệ số Kp, Ki, Kd được lưu trữ trong các thanh ghi 16 bit riêng biệt dưới dạng số bù hai. Các thanh ghi Kp, Ki, Kd, SP được cập nhật sau khi reset hệ thống và sau đó tính Kpd = Kp + Kd và lưu vào thanh ghi Kpd. Tiếp theo hệ thống tính toán với dữ liệu đầu vào là SP và PV. Việc tính toán được hiện trong 7 chu kỳ tương ứng với các trạng thái từ S_CAL_1 đến S_CAL_7. Lưu đồ trạng thái của bộ điều khiển của hệ thống được minh họa ở Hình 4. Trong quá trình tính toán nếu chưa tìm được kết quả u(n) thì các thanh ghi này sẽ không được phép cập nhật bởi vì các giá trị này được sử dụng xuyên suốt quá trình tính toán.



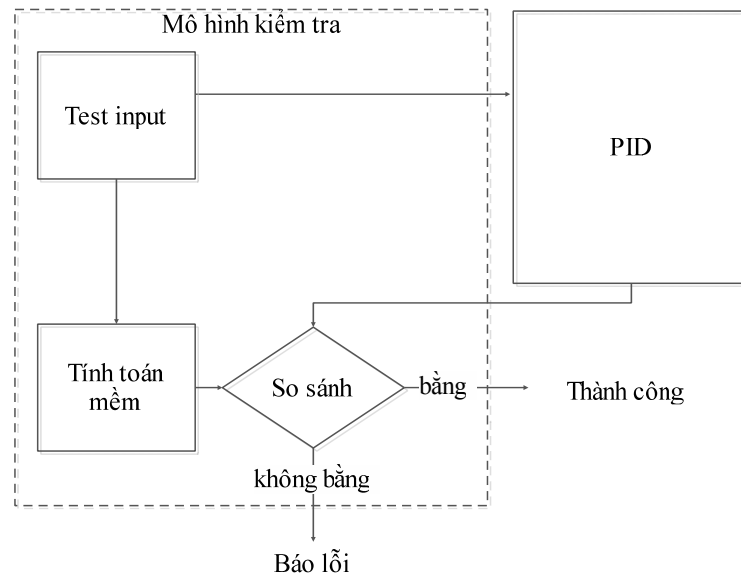
Hình 4. Trạng thái hoạt động của thiết chế PID

Bảng 2. Trạng thái hoạt động của bộ công và bộ nhân

Trạng thái	Mô tả	Hoạt động bộ công		Hoạt động bộ nhân	
		Tầng 1	Tầng 2	Tầng 1	Tầng 2
S_IDLE	Nghỉ				
S_LOAD_K	Lưu các hệ số Kp, Ki, Kd	A= Kp, B=Kd			
S_CAL_KPD	Tính Kp+Kd		Kpd = Kp + Kd		
S_CAL_1		A= SP, B=~PV, Carry=1			
S_CAL_2	Tính e(n)= ERR0= SP-PV	A= ~ERR0, B=0, Carry = 1	Sum=SP+ (~PV)+1, ERR0= Sum	MD=Sum, MR= KI	
S_CAL_3	Tính e(n)*Ki và ERR1= -e(n-1)		Sum=(~ERR0)+1+0, ERR1= Sum	MD= ERR0, MR=KPD	Product =ERR0*KI
S_CAL_4	Tính e(n)*Kpd	A= Product, B= SIGMA		MD= ERR1, MR=KD	Product=ERR0*KPD
S_CAL_5	Tính -e(n-1)* Kd	A= Product, B= Sum	Sum=Product+ SIGMA, SIGMA=Sum		Product=ERR1*KD
S_CAL_6	Tính Sigma=e(n)*Ki+Sigma trước đó	A= Product, B= Sum	Sum=ERR0*KPD +SIGMA		
S_CAL_7	Tính e(n)*Kpd+Sigma cập nhật		Sum=ERR0*KPD+SIGMA+ ERR1*KPD, UN=Sum		
S_FINISH	Xuất kết quả				

Kiểm tra hoạt động

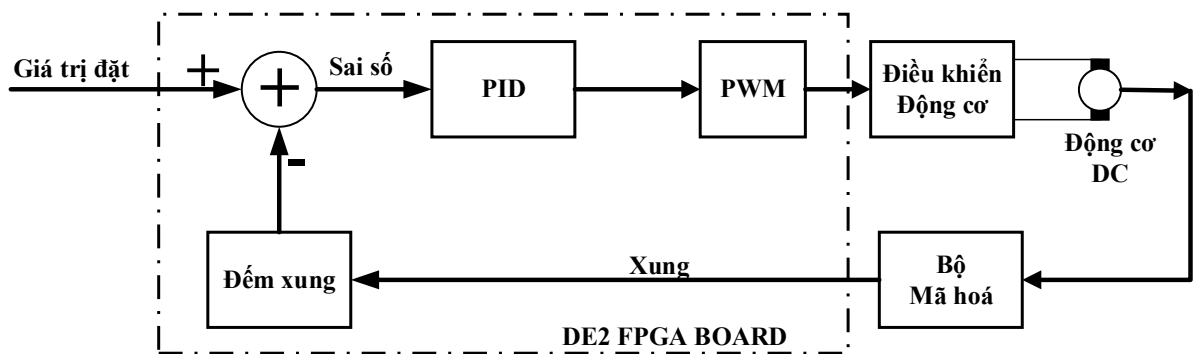
Để kiểm tra hoạt động của hệ thống, chúng tôi xây dựng mô hình kiểm tra bằng verilog. Mô hình được trình bày trong Hình 5. Các đầu vào đưa vào hệ thống PID đồng thời được tính toán trực tiếp trên mô hình và ghi các giá trị vào thanh ghi kết quả. Khi PID tính toán xong giá trị sẽ được so sánh với kết quả mô hình. Nếu có sự sai khác sẽ thông báo lỗi. Hệ thống kiểm tra và cho kết quả chính xác hoàn toàn.



Hình 5. Mô hình kiểm tra thiết kế PID

Thiết kế được tổng hợp trên FPGA Cyclone II. Kết quả tổng hợp cho lõi PID đạt tần số hoạt động tối đa là 148 MHz, tổng số logic element là 1040.

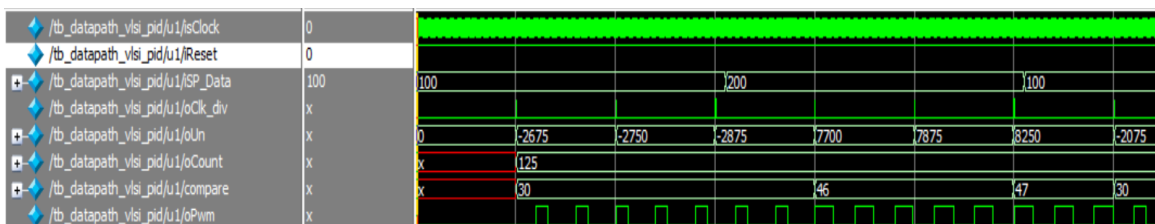
IV. THỰC NGHIỆM ĐIỀU KHIỂN ĐỘNG CƠ DC



Hình 6. Mô hình kiểm soát tốc độ động cơ DC trên FPGA

Sau khi thiết kế thành công bộ điều khiển PID trên FPGA, chúng tôi ứng dụng bộ điều khiển PID này vào mô hình kiểm soát tốc độ động cơ DC. Mô hình này được trình bày trong Hình 6. Mô hình được hiện thực trên FPGA kết hợp với một số mạch bên ngoài, gồm có các khối dưới đây.

- ✓ Bộ điều khiển PID theo cấu trúc trình bày ở phần trên. Tín hiệu vào Set Point, 16 bit, của PID là hiệu số giữa số xung xác lập và số xung đếm được trong 100 ms (tương đương vận tốc quay của động cơ DC), tín hiệu ra, 32 bit có dấu, là duty cycle cho bộ tạo xung PWM
- ✓ Bộ tạo xung PWM nhằm điều chỉnh tốc độ quay động cơ DC. Tín hiệu ra là xung PWM tần số 80 Hz với duty cycle theo bộ điều khiển PID
- ✓ Mạch điều khiển động cơ là một mạch cầu H, bộ khuếch đại công suất điều khiển động cơ, kết nối bộ tạo xung PWM,
- ✓ Mạch mã hoá chuyển đổi tốc độ quay của động cơ DC thành xung số (11 xung/vòng)
- ✓ Bộ đếm xung (pulse counter) đếm xung trong khoảng thời gian $\Delta t = 100$ ms



Hình 7. Dạng sóng mô phỏng hệ thống trên Modelsim

Bảng 3. Mô tả hoạt động của hệ thống kiểm soát tốc độ động cơ

Trạng thái	Mô tả
T1	Thiết lập vận tốc quay
T2	So sánh vận tốc quay thiết lập với vận tốc quay thực tế $Error = SetPoint - PulseCounter$ (xung)
T3	Tính PID, kết quả duty cycle của PWM
T4	Tạo xung PWM theo duty cycle của PID với tần số 80 Hz
T5	Đo vận tốc quay bằng cách đếm xung trong khoảng thời gian 100ms
T6	Lặp lại từ T2

Hình 7 là dạng sóng mô phỏng hệ thống trên Modelsim theo qui trình hoạt động mô tả trong bảng 3. Với tần số mô phỏng 50MHz, trạng thái thiết lập số vòng quay, SP_Data, lần lượt thay đổi 100, 200, 100 vòng để kiểm tra đáp ứng của hệ thống; sau khoảng thời gian 100ms, bộ đếm kết thúc hoạt động bằng tín hiệu xung *oCLK_div* và bộ PID thực hiện tính toán trong 8 chu kỳ xung hệ thống để điều khiển duty cycle, *tín hiệu compare*, cho bộ PWM thể hiện qua *oPwm*.

Bảng 4. So sánh kết quả nghiên cứu với các thiết kế lỗi PID khác

Thiết kế lỗi PID	[1]	[3]	Đề suất
Số bit ngõ vào	12	12	16
Số bit ngõ ra	8	26	32
Số bộ nhân	2	3	1
Số bộ cộng	3	5	1
Tốc độ lỗi PID			148 MHz
Tốc độ tổng hợp toàn hệ thống			128 MHz
Tốc độ thử nghiệm toàn hệ thống		50 MHz	50 MHz

- Kết quả tổng hợp toàn hệ thống, ở hình 6, trên FPGA, đạt tần số 128 MHz với tổng số logic element là 2114.

Nghiên cứu đề suất lỗi PID chỉ sử dụng 1 bộ cộng, 1 bộ nhân, xử lý pipeline 2 tầng; trong khi thiết kế [1] cần 2 bộ nhân, 3 bộ cộng; thiết kế [3] cần 5 bộ nhân, 3 bộ cộng và cả 2 thiết kế [1], [3] đều xử lý song song kết hợp tuần tự.

V. KẾT LUẬN

Bộ PID đa tầng pipeline tốc độ cao thực hiện đối với các hệ số đầu vào K_p , K_i , K_d , SP và PV là số 16 bit có dấu, ngõ ra $u(n)$ là số 32 bit. Hệ thống chỉ sử dụng 1 bộ cộng Carry Look Ahead và 1 bộ nhân nhanh Booth multiplier, pipeline 2 tầng cho toàn bộ tác vụ do đó tiết kiệm tài nguyên và công suất. Bộ PID đã đáp ứng được mô hình điều khiển vận tốc động cơ DC theo thời gian thực, được tổng hợp trên FPGA Cyclone 2 của Altera. Tần số hoạt động của hệ thống tổng hợp trên FPGA Cyclone 2 của Altera 128 MHz. Thiết kế PID trong bài báo này theo hướng VLSI, tiền đề cho việc thiết kế vi mạch kết hợp PID với các thuật toán chọn lựa hệ số thông minh hơn.

Nghiên cứu được tài trợ bởi Đại học Quốc gia Thành phố Hồ Chí Minh (ĐHQG-HCM) trong khuôn khổ Đề tài mã số C2017-18-05/HĐ-KHCN.

TÀI LIỆU THAM KHẢO

- [1] Michal Kocur, Stefan Kozak, Branislav Dvorscak, "Design and Implementation of FPGA - Digital Based PID Controller", IEEE, 2014.
- [2] Vipul B. Patel, Virendra Singh, Ravi H. Acharya, "Design of FPGA-based All Digital PID Controller for Dynamic Systems", International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 1, Issue 2, August 2012.
- [3] Şirin Akkaya, Onur Akbatı, Haluk Görgün, "Multiple Closed Loop System Control with Digital PID Controller Using FPGA", IEEE, 2014.
- [4] Basil Hamed, Moayed Al-Mobaied, "Fuzzy PID Controllers Using FPGA Technique for Real Time DC Motor Speed Control", Intelligent Control and Automation, 2011.

HIGH SPEED MULTI PIPELINE PID CONTROLLER FOR DC MOTOR ON FPGA

Bui An Dong, Huynh Quoc Thinh, Bui Trong Tu

Viet Nam National University of Science, Ho Chi Minh City

ABSTRACT: *This paper presents the multi pipeline high speed VLSI design of PID (Proportional-Integral-Derivative) algorithm and the real time DC motor speed control system on FPGA. The authors propose the ASIC (Application Specific Integrated Circuit) of PID, includes 16-bit inputs (K_p , K_i , K_d and set point), Booth multiplier, Carry Look Ahead and pipelined processing and 32-bit output. This PID design was synthesized on the FPGA Cyclone II at 148MHz and implemented successfully in real time this FPGA to control DC motor speed with frequency 128MHz.*