Notice of Violation of IEEE Publication Principles

"FPGA Based Efficient Implementation of PID Control Algorithm"
by M. Sultan, M. Siddiqui, A.H. Sajid, D.G. Chougule
in the Proceedings of the International Conference on Control, Automation, Communication and Energy Conservation, June 2009

After careful and considered review of the content and authorship of this paper by a duly constituted expert committee, this paper has been found to be in violation of IEEE's Publication Principles.

This paper contains significant portions of original text from the paper cited below. The original text was copied without attribution (including appropriate references to the original author(s) and/or paper title) and without permission. The misconduct was committed specifically by A. H. Sajid (Second author), and his actions were done without the knowledge or approval of M. Sultan M. Siddiqui (First author) and D. G. Chougule (third author).

Due to the nature of this violation, reasonable effort should be made to remove all past references to this paper, and future references should be made to the following article:

“Design and Implementation of Modular FPGA-based PID Controllers”,
by Y.F. Chan, M. Moallem, and W.Wang,
in the IEEE Transactions on Industrial Electronics, vol.54, no.4, Aug. 2007, pp. 1898–1906

FPGA Based Efficient Implementation of PID Control Algorithm

M. Sultan. M. Siddiqui, A.H.Sajid and D.G.Chougule

Abstract— In this paper, an efficient design scheme for implementation of PID control algorithm in FPGA is presented. The algorithm is implemented using a Distributed Arithmetic (DA) based scheme where a Look-Up-Table (LUT) mechanism inside the FPGA is utilized. It can be shown that DA based PID controller saves 80% hardware utilization and 40% savings in power consumption compared to the multiplier-based scheme. It also offers good closed loop performance while using less resources, resulting in cost reduction, high speed and low power consumption which are desirable in embedded control applications. Finally as a case study we discuss the DA based scheme to design a temperature control system. The design uses a modular approach, so that some modules can be reused in other applications. The complete digital control system is built using commercial FPGA's to demonstrate its efficiency. The same approach can be extended to design other embedded controllers using FPGA.

Index Terms—FPGA Design, PID Controller, Distributed Arithmetic (DA), Power and Speed Optimization.

1 INTRODUCTION

THE proportional-integral-derivative (PID) controller is is one of the most common type of feedback controllers lers that are used in dynamic systems [1]. An important tant feature of this controller is that it does not need a precise analytical model of the system that is being controlled. This controller has being widely used in many different areas, such as process control, aerospace, robotics, automation, manufacturing, and transportation system. Implementation of PID controllers has gone through several stages of evolution from early mechanical and pneumatic design to microprocessor-based system. Recently, FPGAs have become an alternative solution for the realization of digital control systems, which were previously dominated by general-purpose microprocessor systems [1], [2]. FPGA technology is now considered by an increasing number of designers in various fields of applications such as wired and wireless telecommunications, image and signal processing, medical equipments, robotics, automotive, and space and aircraft embedded control systems [3].For these embedded applications, reduction of the power consumption, thermal management and packing, reliability and protection against solar radiation are of prime importance [3]. FPGA-based controllers offer advantages such as high speed, complex functionality, low power consumption and reduction in cost. These are attractive features from the embedded system point of view [4]. Another advantage of FPGA-based platforms is their capability to execute concurrent operations, allowing parallel architectural design of digital controllers [5],[6].

Conventional implementations of FPGA-based controllers have not focused on optimal used of hardware resources. These designs usually require a large number of multipliers and adders and don't efficiently utilize the memory rich characteristics of FPGA's. An FPGA chip consists of many memory blocks referred to as LUT's and can be utilized to improve the performance of certain operations such as multiplication, while the tradeoff for speed can be tolerated. In this paper, we study the design and implementation of an efficient PID controller using the DA scheme, which is an efficient LUT design method and is very promising in the FPGA implementation of PID controller. The proposed PID controller reduces the cost of the FPGA design by enabling the chip to accommodate more logic and arithmetic functions while requiring less power consumption compare to the multiplier-based PID controller [7]. In addition due to the flexibility of using LUT's in FPGA's, the design method can be used to implement other algorithms, such as anti-windup compensation or adaptive control schemes.

In this paper, a case study is presented in which a modular FPGA-based design approach is applied to design a temperature control system. The same approach can be extended to design other embedded controllers using FPGA. The complete system is implemented by dividing system functions in reconfigurable module. The organization of the paper is as follows: In section II an improved PID controller is considered and its implementation using DA scheme is discussed. In section III an overview of the components of a general purpose PID based feedback control system is presented followed by an approach for designing the control system using FPGA technology. In section IV the implementation results on a Xilinx FPGA chip are discussed. Comparisons are made between the proposed scheme and the design based on conventional methods. Conclusions are discussed in section V.

2 PID CONTROLLER IMPLEMENTATION USING DA

The application of a PID controller in a feedback control system is shown in Figure 1, where u_c is the command signal, y is the feedback signal, e is the error signal and u is the control output (controller output). The simplest form of PID control algorithm is given by

$$u(t) = K_P e + K_I \int_0^t e(\tau) d\tau + K_D \dot{e} \qquad \text{---- (1)}$$

From a practical point of view, implementation of the above algorithm has certain limitations [8]. Firstly, actuator saturation can cause integrator windup, leading to a sluggish

• M. Sultan M. Siddiqui is with the Department of Electrical Engg., IIT, Delhi, E-mail: msultanms@gmail.com.
• A. H. Sajid is with the Department of Electronics Technology, Shivaji University, Kolhapur. E-mail: ahsajid@yahoo.co.in.
• D. G. Chougule is with the Department of Electronics Engineering, TKIET, Warnanagar.

Manuscript received (21-02-2009)..

transient response. Secondly, the pure differentiation term amplifies noise, leading to a deterioration of the control command. Finally, the differentiation term acts on the error signal, taking the derivative of the command signal as well. This may lead to spikes in the command signal when the user changes the set point abruptly. Further it was found advantageous, not to let the derivative act on the

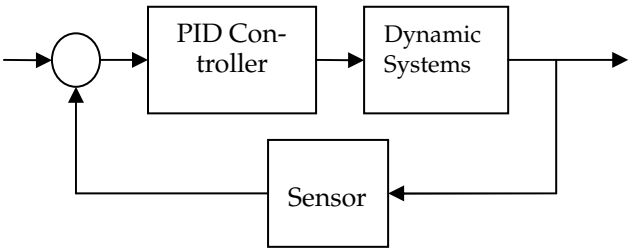

Figure 1. A PID-based feedback control system

command signal and let only a fraction b of the command signal act on the proportional part. The PID algorithm then becomes

$$U(s) = K(bU_c(s) - Y(s) + \frac{1}{sT_i}(U_c(s) - Y(s)) - \frac{sT_d}{(1+\frac{sT_d}{N})}Y(s)) \quad \text{---- (2)}$$

which overcomes the above problems and is given in Laplace domain, where K, b, Ti, Td and N are controller parameters and U(s), U_c (s), and Y(s) denote the Laplace transforms of u, u_c, and y, respectively.

In the FPGA implementation of the PID controller, major effort is placed on the hardware optimization of the controller. In this regard, an area-efficient DA-based algorithm for the PID controller is proposed in this section. An area-efficient controller means that it can fit in a smaller FPGA chip, resulting in cost reduction of the controller hardware. In order to implement the control algorithm using FPGA, (2) has to be discretized. Denoting the sampling period as T and using backward differences to discretize the derivative term and forward differences for the integral term, (2) can be written as

$$u(kT) = P(kT) + I(kT) + D(kT) \quad \text{---- (3)}$$

where 'k' denotes the k^{th} sampling instant, and

$$P(kT) = K(bu_c(kT) - y(kT)) \quad \text{---- (4a)}$$

$$I(kT) = I((k-1)T) + \frac{KT}{T_i}(u_c((k-1)T) - y((k-1)T)) \quad \text{---- (4b)}$$

$$-D(kT) = \frac{T_d}{T_d + NT}D((k-1)T) - \frac{KT_dN}{T_d + NT}(y(kT) - y((k-1)T)) \quad \text{---- (4c)}$$

where y(kT) is the feedback signal at the current instant; y((k - 1)T) is the feedback signal at the previous instant; u_c (kT) is the command signal at current instant and u_c ((k - 1)T) is the command signal at previous instant; I(k - 1) is the integral term at the previous instant; D(k - 1) is the derivative term at the previous instant; K, b, T_i, T_d, N are controller parameters and T is the sampling period. Having obtained the discretized control algorithm, the focus is then placed on its efficient implementation. The direct implementation of the terms in (4(a),(b), & (c)) using FPGA requires a total of 5 multipliers, 7 adders/substractors, and 4 delay blocks. The P(kT) term requires 2 multipliers and 1 adder/ substractor, I(kT) requires 1 multiplier and 2 adders/ substractors, D(kT) requires 2 multipliers and 2 adders/ substractors, and u(kT) requires 2 adders. Each of u_c (k - 1), y(k - 1), I(k - 1), and D(k - 1) terms in equation 4 requires 1 delay block. Thus, a total of 4 delay blocks are required.

The multiplier-based design uses many multipliers and adders. Since FPGA has a limited number of CLB's, for the above calculation, the multiplier-based design is not efficient for FPGA. In order to reduce the required multiplier and adders, we apply the DA method [9] in order to replace the multiplication operation by simple shifting and addition operation. The procedure is discussed in the following

1) DA: DA [9] is a bit-serial computation algorithm that performs multiplication using an LUT-based scheme. Consider a sum of product calculation that is written as

$$Y = \sum_{i=0}^{N-1} A_i x_i \quad \text{---- (5)}$$

Where A_i are constant coefficients, x_i's are input data of size N and y is the output. Let represent x_i in bitwise format and assuming that it is a 2's compliment fractional number; we have [10],

$$x_i = -x_{i,0} + \sum_{j=0}^{M-1} x_{i,j} * 2^{-j} \quad \text{---- (6)}$$

where x_{i0} is sign bit, $x_{i,j}$ is the j^{th} bit of x_i having value 0 or 1, $x_{i,\,M-1}$ is the LSB and M is the word size. Substituting equation 6 in equation 3, we get

$$Y = \sum_{i=0}^{N-1} A_i(-x_{i,0} + \sum_{j=1}^{M-1} x_{i,j} * 2^{-j})$$

$$Y = -\sum_{i=0}^{N-1} A_i x_{i,0} + \sum_{i=0}^{N-1} A_i(\sum_{j=1}^{M-1} x_{i,j} * 2^{-j})) \quad \text{---- (7)}$$

Equation 7 is the conventional form of expressing the inner product. Direct mechanization of this equation defines "lumped" arithmetic computation. Let us interchange the order of summations, which gives us

$$Y = \sum_{j=1}^{M-1}(\sum_{i=0}^{N-1} A_i x_{i,j}) * 2^{-j} - \sum_{i=0}^{N-1} A_i x_{i,0} \quad \text{---- (8)}$$

Equation 8 defines a distributed arithmetic computation. Consider the second term of equation 8 and define it as

$$Z_0 = -\sum_{i=0}^{N-1} A_i x_{i,0} \quad \forall j = 0 \quad \text{---- (9a)}$$

and the first term of the equation as

$$Z_i = \sum_{i=0}^{N-1} A_i x_{i,j} \quad \forall j \neq 0 \quad \text{---- (9b)}$$

Since $x_{i,j}$ can take 0 or 1 value only, equation 9b can have only 2^N possible values for Z_j. Therefore, they can be pre-computed and stored in an LUT. Each x_i, where I = 0 to N - 1, forms the address of the LUT. The output Y can then be expressed as follows

$$Y = \sum_{j=0}^{M-1} Z_j * 2^{-j} \quad \text{---- (10)}$$

The result Y can then be computed using shifting and addition operations for all Z_j's from j^{th} bit.

2) DA-Based PID Controller: Let us consider the controller terms given in equation 4(a),(b), and (c). Assuming that u_c(kT), u_c ((k - 1)T), y(kT) and y((k - 1)T) are M bit numbers and j represents the j^{th} bit of the numbers, we have

$$P(kT) = \sum_{j=0}^{M-1} K(bu_c(kT)(j) - y(kT)(j)) * 2^j \quad \text{---- (11)}$$

$$I(kT) = \sum_{j=0}^{M-1} (I((k-1)T)(j) + \frac{KT}{T_i} u((k-1)T)(j) - y((k-1)T)(j)) * 2^j \quad \text{----- (12)}$$

$$D(kT) = \sum_{j=0}^{M-1} (\frac{T_d}{T_d+NT} D((k-1)T)(j) - \frac{KT_d N}{T_d+NT} ((y(kT)(j) - y((k-1)T)(j)) * 2^j)) \quad \text{---- (13)}$$

The results of (Kb u_c (kT)[j] – K y(kT)[j], [I ((k - 1)T)[j] + (KT/T_i)[u_c ((k –1)T)[j] - y((k - 1)T)[j]], and ((T_d / T_d + NT) D ((k - 1)T)[j] - (KT_d N/ T_d + NT) ((y (kT)[j] - y ((k - 1)T) [j])) can be precomputed and stored in 3 LUT's namely, LUT_P, LUT_I, and LUT_D. The contents of the 3 LUT's are shown in Tables I, II and III, respectively. Using the 3 LUT's and the corresponding shift-add accumulators (ACCs), the P(kT), I(kT), and D(kT) terms can be obtained in m clock cycles. The main advantage of DA expression given by equations 11, 12, and13 lies in its capability to compute the PID function utilizing the LUT-rich FPGA.

TABLE I
Contents of LUT_P

u_c (kT)[j]	y(kT)[j]	LUT_P
0	0	0
0	1	- K
1	0	K b
1	1	Kb – K

TABLE II
Contents of LUT_I

I((k-1) T) [j]	u_c((k- 1)T)[j]	y((k -1)T)[j]	LUT_I
0	0	0	0
0	0	1	- (KT/T_i)
0	1	0	KT/T_i
0	1	1	0
1	0	0	1
1	0	1	1-(KT/ T_i)
1	1	0	1+(KT/T_i)
1	1	1	1

TABLE III
Contents of LUT_D

D((k-1)T) [j]	y(kT) [j]	y((k-1)T)[j]	LUT_D
0	0	0	0
0	0	1	$KT_dN/(T_d+NT)$
0	1	0	$-(KT_dN/(T_d+NT))$
0	1	1	0
1	0	0	$T_d/(T_d+NT)$
1	0	1	$(1+KN)T_d/(T_d+NT)$
1	1	0	$(1-KN)T_d/(T_d+NT)$
1	1	1	$T_d/(T_d+NT)$

Based on the above equations, the DA implementation of the PID controller is shown in figure 2. It consists of 4 delay blocks, 3 LUTs, 3 ACCs, and 2 adders. The delay blocks 1 and 2 are used to obtain u((k – 1)t) and y((k – 1)T), respectively. The delay blocks 3 and 4 are used to generate the terms I((k – 1)T) and D((k – 1)T), respectively. 3 LUTs and 3 ACCs are used to provide the term P(kT), I(kT), and D(kT). The throughput of this PID implementation is (m + 1) clock cycles, i.e., m clock cycles to generate the result and one more clock cycle to update the I((k – 1)T) and D((k – 1)T) terms. For the multiplier-based controller throughput is 1 clock cycle. The latency is also (m + 1) clock cycles, where as for multiplier-based method, it is 1 clock cycle.

3 FPGA Implementation Results

The proposed DA-based PID controller is implemented using the Xilinx Inc. FPGA technology and can be used as a general purpose controller for different applications. The FPGA design flow is as follows. First, the controller was implemented by using the Xilinx ISE foundation tools [11] and simulated at the Register Transfer Level (RTL) using Modelsim tool to verify the correctness of the design. By using the Xilinx ISE foundation tools, the logic synthesis was carried out to optimize the design, and the placement and routing were carried out automatically to generate the FPGA implementation file. Finally, the generated implementation file was downloaded to the FPGA development board for testing. This PID controller is targeted to a Xilinx Spartan II-E FPGA for a 13 bit input. ACC

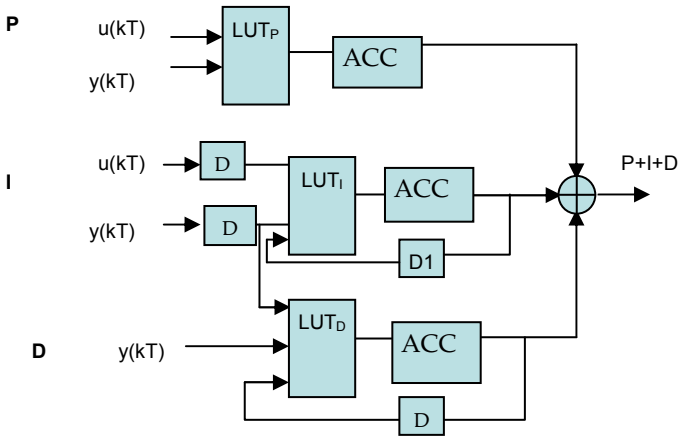

Figure 2. Architecture of DA-Based PID controller

For the purpose of comparison, both multiplier-based and DA-based PID controllers are implemented. Due to the bit-serial nature of the DA-based PID controller, parallel-in-serial-out shift registers are used to serialize the inputs. The

implementation results of both the designs were shown in table IV. The clock frequency of both designs is 50 MHz.

TABLE IV
FPGA Implementation Results of PID controller

PID Controller	complexity	Throughput	latency	power
DA-Based	3 LUTs, 3 ACCs, 4 delay blocks, 2 adders	m+1 clock cycles	m+1 clock cycles	9.29mW
Multiplier-Based	5 MULs, 4 delay blocks, 5 adders	1 clock cycle	1 clock cycle	15mW

It is seen from Table IV that the DA design uses about 13% of the logic resources required by the design using multipliers. On the other hand, the power consumption of DA-based design is reduced by 40%. Due to the serial nature of the DA method, the DA PID controller needs 14 clock cycles, while the design using multipliers needs 1 clock cycle. Since in the control system considered, the 14 cycles of 50 MHz clock are fast enough, the tradeoff of speed to hardware resource and power saving is useful for the PID controller design. A design which is efficient in terms of power consumption and chip area, means that the FPGA chip can be used to accommodate more controllers with adequate speed and low power consumption, resulting in a cost reduction of the controller hardware.

4 Application to Temperature Control System

The block diagram of a general purpose PID based feedback system is shown in Figure 3, where uc is the command signal, y is the feedback signal, e is the error signal, and u is the control input. Figure 4 shows the block diagram of FPGA-based temperature control system. The system consists of following components: 1) a tube with a fan, a light bulb, and a thermistor; 2) an I/O panel that includes a Digilent DIO2 board, push-button keys with an LCD display; 3) an ADC chip; 4) FPGA development board consisting a Xilinx Spartan-2E FPGA.

The thermistor is used for temperature measurement inside the tube, output of thermistor i.e. voltage across thermistor, is used to calculate the temperature that is sampled by the ADC to be used in the control law. Two PWM generators are implemented to control the dc motor and the lamp. The motor and the lamp are turned either ON or OFF depending on the PID controller output u. If u is positive, the opposite will be true. In our implementation, u is a 34-bit fixed-point two's complement number. It is converted to a sign and magnitude representation, and then scaled accordingly in order to generate the correct PWM output. Both PWM generators run at a clock frequency of 50 MHz with a PWM period of 20 ms The ADC that is used in this system is a 12-bit ADC with an internal reference voltage of 4.096 V. The ADC Interface polls the ADC every 20 ms to receive the sampled data. A PS2 keyboard, which is connected to the I/O board, is used for user input. An LCD on the I/O board displays the user's requests that are entered from the keyboard. The User Interface decodes the user's requests and sends appropriate commands to other modules in the system.

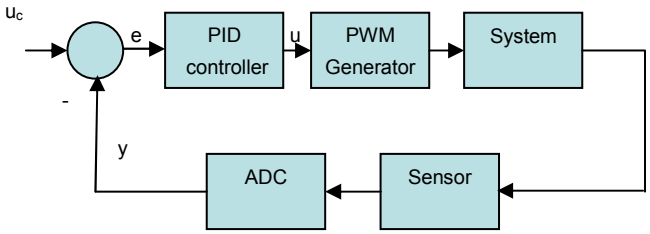

Figure 3. Block Diagram of a general PID-based feedback control system.

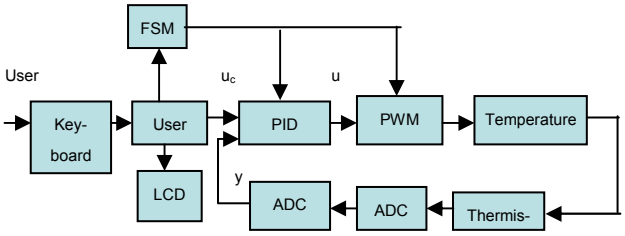

Figure 4. Block Diagram of FPGA-Based Temperature control system.

5 Conclusion

In this paper, a novel DA-based PID controller was presented, for FPGA implementation. By using the DA- based LUT scheme, the memory inside FPGA has been utilized to provide efficient design for PID controllers, resulting in reduction of FPGA design cost. In addition, due to the flexibility of the LUT in the FPGA, this FPGA-based PID controller can be easily extended to incorporate other algorithms, such as antiwindup protection or adaptive scheme. The FPGA implementation results shows that, the DA design requires only 13% of logic devices compared to the design using multipliers. Further more, the power consumption is reduced by about 40%.

References

[1] K. J. Astrom and B. Wittenmark, *Computer Controlled Systems* .Englewood Cliffs, NJ: Prentice-Hall, 1997.

[2] L. Samet, N. Masmoudi, M. W. Kharrat, and L. Kamoun, "A digital PID controller for real-time and multi-loop control: A comparative study," in *Proc. IEEE Int. Conf. Electron., Circuits and Syst.*, Sep. 1998, vol. 1,pp. 291–296.

[3] E. Monmasson and M. N. Cirstea: "*FPGA Design Methodology for Industrial control systems- A Review"*, IEEE Transaction on Industrial Electronics vol 54, no 4, Aug 207 pp 1824-1841.

[4] W.Wolf, *Computers as Components: Principles of Embedded Computing System Design*. San Francisco, CA: Morgan Kaufmann, 2001.

[5] P. Zumel, A. de Castro, O. Garcia, T. Riesgo, and J. Uceda, "Concurrent and simple digital controller of an AC/DC converter with power factor correction," in *Proc. IEEE Appl. Power Electron. Conf. Expo.*, 2002, vol. 1, pp. 469–475.

[6] L. Charaabi, E. Monmasson, and I. Slama-Belkhodja, "Presentation of an efficient design methodology for FPGA implementation of control systems: Application to the design of an antiwindup PI controller," in *Proc. IEEE Ind. Electron. Soc. Annu. Conf.*, Nov. 2002, vol. 3, pp. 1942–1947.

[7} F. Ricci and H. Le-Huy, "An FPGA-based rapid prototyping platform for variable-speed drives," in *Proc. IEEE Ind. Electron. Soc. Annu. Conf.*, Nov. 2002, vol. 2, pp. 1156–1161.

[8] B. Wittenmark, K. J. Astrom, and K.-E. Arzenin "Computer control: An overview," Dept. Autom. Control, Lund Inst. Technol., Lund, Sweden, Apr. 2003, Tech. Rep. [Online]. Available: www.control.lth.se/kursdr/ifac.pdf

[9] A. White, "Application of distributed arithmetic to digital signal processing: A tutorial review," *IEEE Trans. Acoust., Speech, Signal Process. Mag.*, vol. 6, no. 3, pp. 4–19, Jul. 1989.

[10] U.. Meyer-Baese "*DSP with FPGAs*" 2nd Edition, Springer, 2004.

[11] www.xilinx.com

M. Sultan M. Siddiqui received his BE in Electronics and Communication Engineering from Bangalore University, Bangalore and M.Tech. in Digital Systems and Computer Electronics from JNT University, Hyderabad. He is presently a research scholar at the Department of Electrical Engineering, Indian Institute of Technology, Delhi, New Delhi and is pursuing Ph.D. studies in computational neuroscience**.**

A. H. Sajid received his BE in Electronics and Instrumentation Engineering from Gulbarga University, Gulbarga in 2001.He Joined as Lecturer in Shadan College of Engineering and Technology, Hyderabad.Presently he is pursuing his M.Tech in Electronics Technology from Shivaji University, Kolhapur. His area of intrest includes FPGA Based System Design, CMOS VLSI Design, and Embedded Systems.

D. G. Chougule received his BE in Electronics Engineering from Shivaji University in1987 and M.E in Electronics Engineering from Walchand College of Engineering in 1992. He joined T.K.I.E.T. Warananagar as Assistant Professor in Electronics Engineering in 1993. At present he is working as Head of Department.He is also a visiting Prof. for M.Tech. in Electronics Technology, Shivaji University Kolhapur.His area of interest includes Digital Signal Processing, Speech, & Image Processing. He guided 15 M.E./M.Tech. Projects