

Projeto e *Layout* de um Circuito Analógico Gerador de Funções de Pertinência Fuzzy tipo-2

Gabriel A. F. Souza¹, Rodrigo B. Santos¹, Paloma M. S. Rocha-Rizol², Lester A. Faria¹

¹Instituto Tecnológico de Aeronáutica - ITA, São José dos Campos/SP,

²Universidade Estadual Paulista - UNESP, Guaratinguetá/SP

Resumo — Este artigo apresenta uma nova arquitetura para um circuito analógico gerador de funções de pertinência fuzzy tipo-2 intervalar em modo corrente. A utilização de um circuito deslocador de corrente permite a geração simultânea das funções de pertinência superior e inferior. Três correntes de programação definem a forma da função de pertinência – trapezoidal, triangular, S ou Z – e uma entrada de tensão define o tamanho da “mancha de incerteza”. A fim de se fabricar o dispositivo em Circuito Integrado, após as devidas simulações e testes, foi desenvolvido o *layout* da arquitetura proposta em tecnologia TSMC 0.18 μ m. O presente artigo realiza a análise das simulações pré e pós-*layout*, comparando resultados decorrentes do posicionamento das estruturas no CI e inferindo eventuais problemas e correções que tal *layout* inseriu no circuito.

Palavras-Chave — Fuzzy tipo-2, Circuitos Analógicos, CMOS.

I. INTRODUÇÃO

O circuito gerador de funções de pertinência – também chamado de fuzificador – é o primeiro bloco de um controlador baseado na lógica *fuzzy*. Neste tipo de controlador, o sinal de entrada é mapeado em conjuntos que associam intervalos do universo de discurso a um “grau de pertinência”. Enquanto na lógica booleana são atribuídos apenas os valores lógicos “verdadeiro” ou “falso” ao pertencimento a um conjunto, na lógica *fuzzy* o “grau de pertinência” pode assumir qualquer valor entre 0 e 1 [1]. À função que associa os pontos do universo de discurso ao “grau de pertinência” dá-se o nome de função de pertinência.

Apesar do nome, a lógica *Fuzzy* é determinística e trabalha com conjuntos exatos, os quais não levam em consideração possíveis incertezas na determinação do “grau de pertinência”. O tratamento destas incertezas pode ser feito utilizando a lógica *fuzzy* tipo-2, em que a função de pertinência mapeia cada ponto do universo de discurso a um conjunto *fuzzy* – agora denominado *fuzzy* tipo-1 [2]. Se estes conjuntos forem todos uniformes, resulta um conjunto *fuzzy* tipo-2 intervalar, conforme mostra a Fig. 1.

Nesse caso fica evidente que o conjunto intervalar fica bem definido somente pelos conjuntos *fuzzy* tipo-1 que delimitam os limites superior e inferior, sendo a área entre essas duas funções chamada de “mancha de incerteza”. Para exemplificar, considere que, na Fig.1, para uma dada entrada do controlador, a pertinência a um conjunto fuzzy tipo-1 é de 0,4. Em um conjunto *fuzzy* tipo-2 intervalar similar, o grau de pertinência seria o intervalo [0,35;0,45], ou seja,

agora existe uma “margem de erro”, ou incerteza associada. Essa característica reflete melhor a realidade e é o que permite aos controladores que trabalham com a lógica *fuzzy* tipo-2 um potencial de apresentar melhor desempenho quando comparado à lógica *fuzzy* tipo-1 [2].

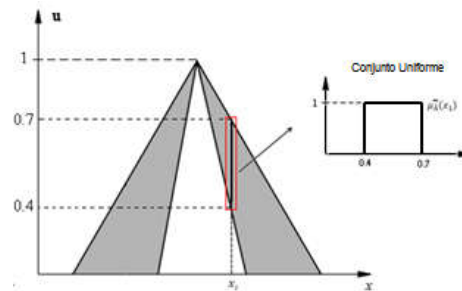


Fig. 1. Representação de um conjunto *Fuzzy* tipo-2 intervalar.

Devido a sua estrutura mais simples e menor custo computacional comparado com o caso geral, os controladores *fuzzy* tipo-2, na maioria das aplicações práticas, utilizam a lógica *fuzzy* tipo-2 intervalar [3]. Um diagrama de blocos que representa um controlador *Fuzzy* tipo-2 é visto na Fig. 2.

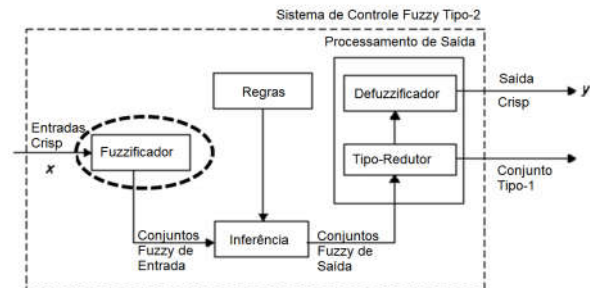


Fig. 2. Diagrama de blocos de um controlador *Fuzzy* tipo-2.

O fuzificador é responsável por fazer a associação da entrada “*crisp*” às funções de pertinência de entrada. Em todos os controladores *fuzzy* aplicados a sistemas reais, a fuzificação é do tipo “*singleton*”, ou seja, o bloco de inferência recebe somente o valor do grau de pertinência da entrada ao conjunto *fuzzy* analisado, e esse valor é o “nível de disparo” utilizado nas regras do bloco de inferência. Como afirmado anteriormente, um conjunto *fuzzy* tipo-2 intervalar é definido somente pela mancha de incerteza. Portanto, basta ao fuzificador determinar os limites superior e inferior dessa mancha. Como o fuzificador é do tipo “*singleton*”, somente os valores do “intervalo de disparo” – ou os “níveis de disparo” inferior e superior – são passados ao bloco de inferência. Este bloco, responsável por fazer a aplicação das regras do tipo “SE – ENTÃO”, utiliza os “níveis de disparo”

recebidos para computar os “níveis de disparo” do antecedente da regra e faz a implicação sobre o conseqüente da mesma. Um único conjunto é obtido fazendo a agregação entre os conjuntos do conseqüente de cada regra após a implicação. O conjunto *fuzzy* tipo-2 intervalar resultante é passado ao bloco tipo-redutor, que o transforma em um conjunto *fuzzy* tipo-1 uniforme. O procedimento de defuzificação então se resume a calcular a média entre os intervalos deste conjunto para determinar o valor de saída “*crisp*” do controlador.

A própria natureza de um conjunto *fuzzy* favorece a utilização de um circuito analógico para a realização do processo de fuzificação. Além dessa, outras vantagens surgem intrinsecamente de uma implementação analógica de tais circuitos como: menor consumo de potência, menor área ocupada e maior velocidade no processamento. Até o momento, os circuitos geradores de função de pertinência tipo-2 analógicos apresentados na literatura são todas combinações de circuitos do tipo-1. O circuito apresentado em [4] funciona em modo corrente com um fuzificador gerando uma função de tipo-1 e um circuito deslocador de corrente que subtrai um valor constante para criar outra função tipo-1. As duas formam as funções superior e inferior do conjunto *fuzzy* tipo-2. Já em [5] são utilizados dois circuitos fuzificadores tipo-1 operando em modo de transcondutância (entrada de tensão e saída de corrente) para criar as duas funções do conjunto tipo-2. Em [6] é utilizado um circuito fuzificador tipo-1 em modo corrente e um circuito deslocador, como em [4], porém com a possibilidade de limitar a corrente da função inferior, alterando sua forma para trapezoidal, mesmo que a função superior seja triangular. Em [7] também são utilizados dois circuitos fuzificadores tipo-1, porém em modo corrente, para criar o conjunto tipo-2. O circuito apresentado neste artigo apresenta como contribuição a utilização de uma arquitetura com espelho deslocador de corrente, a qual é capaz de gerar simultaneamente as funções de pertinência superior e inferior, definindo a mancha de incerteza de um conjunto tipo-2. Tal arquitetura é inédita na literatura e apresenta uma série de vantagens sobre as anteriormente citadas neste artigo (menor consumo de potência, menor área ocupada e maior velocidade no processamento, decorrentes intrinsecamente da implementação analógica do circuito, não sendo, com isso, analisadas nesse trabalho). Após as devidas simulações do circuito projetado, o *layout* da arquitetura proposta foi desenvolvido em tecnologia TSMC 0.18 μ m, proporcionando uma análise dos resultados das simulações pré e pós-*layout*, e uma comparação dos resultados decorrentes do posicionamento das estruturas no CI, inferindo eventuais problemas e correções que tal *layout* inseriu no circuito.

II. CIRCUITO FUZIFICADOR

O circuito em questão funciona em modo corrente e é capaz de gerar funções de pertinência com forma triangular, trapezoidal, S e Z. A forma geral é trapezoidal, com duas correntes de programação que definem os cantos do trapézio (I_{T1} e I_{T2}) e uma corrente definindo o valor máximo, ou “1”, do grau de pertinência (I_{MAX}), conforme mostrado na Fig.3.

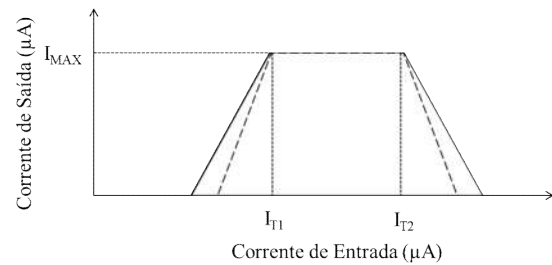


Fig. 3. Forma geral de uma função de pertinência fuzzy tipo-2 intervalar gerada pela arquitetura proposta.

A forma triangular pode ser gerada fazendo $I_{T1} = I_{T2}$, enquanto a forma Z pode ser criada fazendo $I_{T1} = 0$ e a forma S fazendo I_{T2} maior que a corrente máxima válida para o universo de discurso. Uma entrada de tensão possibilita a alteração do tamanho da “mancha de incerteza”, enquanto a inclinação “média” de cada um dos lados é determinada digitalmente por uma palavra de 2 bits.

Cada uma das “pernas” do trapézio é gerada por um espelho deslocador de corrente [8], exibido na Fig. 4.

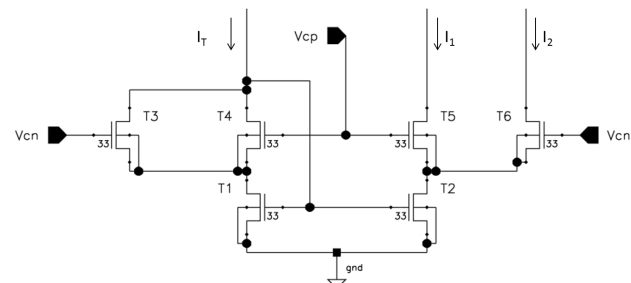


Fig. 4. Circuito do espelho deslocador de corrente.

Este circuito funciona dividindo uma cópia da corrente de entrada em duas partes. A razão entre as correntes de saída é determinada pela diferença entre as tensões de polarização V_{cp} e V_{cn} . Os transistores T2, T5 e T6 foram escolhidos com o dobro da largura dos transistores T1, T3 e T4, de forma que vale a relação $2I_T = I_1 + I_2$. Em outras palavras, quando $V_{cp} = V_{cn}$, então $I_1 = I_2 = I_T$. Quanto maior a diferença entre as tensões $\Delta V_c = V_{cp} - V_{cn}$, maior a diferença entre as correntes I_1 e I_2 , o que corresponde a um aumento da “mancha de incerteza” – ou da diferença entre as correntes de saída superior e inferior.

Para tanto, são utilizados dois espelhos de corrente. Em um deles, I_T é igual à diferença entre a corrente de entrada (I_{IN}) e a corrente de programação I_{T1} . No outro, I_T é dada pela diferença entre I_{T2} e I_{IN} . As correntes I_1 e I_2 de cada um dos espelhos são somadas para formar as correntes superior e inferior. Apesar de os circuitos deslocadores de corrente serem capazes de criar as correntes que formam as laterais esquerda e direita, variando a diferença entre as correntes superior e inferior, a corrente média permanece constante. Dessa forma, um circuito digitalmente programável é utilizado para alterar a inclinação média. Uma palavra de 2 bits (V_1 e V_2) seleciona entre os possíveis ganhos 1, 2, 3 e 4. Para facilitar a ligação com os circuitos deslocadores de

corrente, este circuito foi desenvolvido utilizando transistores PMOS, conforme mostra a Fig. 5.

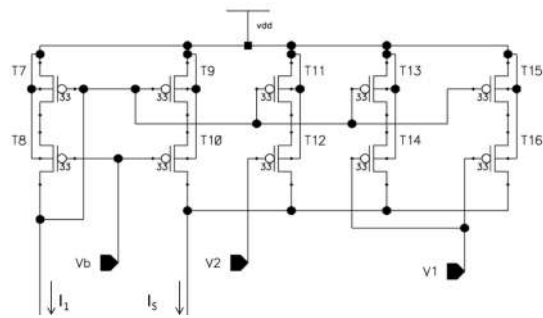


Fig. 5. Circuito do espelho de corrente digitalmente programável.

É possível observar que o circuito funciona simplesmente alterando o número de cópias da corrente original dependendo da tensão aplicada na porta dos transistores T12, T14 e T16. São utilizados dois destes circuitos, um para cada uma das correntes de saída, superior e inferior.

A corrente máxima é definida fazendo a diferença limitada entre I_{MAX} e as correntes de saída dos circuitos digitalmente programáveis, utilizando espelhos de corrente simples em configuração de baixa tensão.

III. LAYOUT

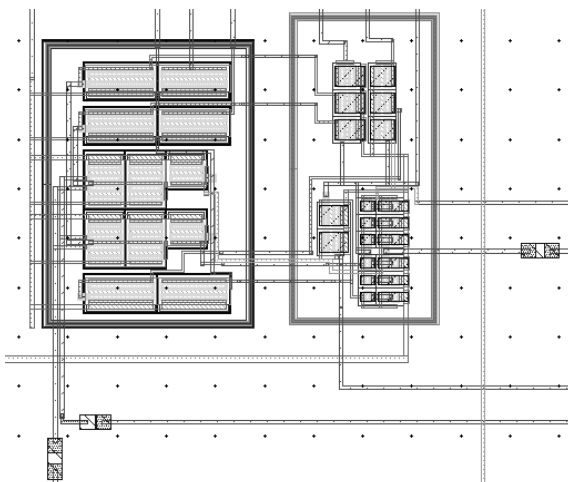


Fig. 6. Layout completo do circuito gerador de funções de pertinência.

O layout do circuito completo, composto por dois espelhos deslocadores de corrente, dois espelhos digitalmente programáveis e dois espelhos de corrente de baixa tensão foi construído utilizando tecnologia TSMC 0.18 μ m, com 6 metais, e está apresentado na Fig. 6. Após a realização das verificações de Design Rule Check (DRC) e Antenna Check – utilizando a ferramenta Assura do software CADENCE – foi constatada a necessidade de incluir “guard rings” ao redor dos transistores NMOS e PMOS para evitar a ocorrência de latch-up, bem como incluir vias passando pelo metal mais alto próximo aos nós conectados diretamente às portas para evitar o “efeito antena”.

O fenômeno de latch-up ocorre quando são formados transistores bipolares de junção parasitas, conectados de tal forma que um ciclo de realimentação positiva pode ocorrer e deixá-los permanentemente ligados enquanto houver alimentação [9]. Para a tecnologia utilizada, as regras de projeto exigem uma distância maior entre os transistores NMOS e PMOS ou a colocação de “guard rings” entre eles. Os “guard rings” são implantes n+ no poço n dos transistores PMOS e p+ no substrato p dos transistores NMOS que fornecem um caminho de baixa resistência para alimentação e terra e reduzem a possibilidade de interferência decorrente dos sinais de outros circuitos [9]. Neste caso optou-se pela criação dos “guard rings” por ser a solução que ocuparia menor área, o que nem sempre é o caso.

O “efeito antena” pode ocorrer durante a fabricação do circuito integrado quando cargas se acumulam em uma área grande de metal conectada diretamente a porta de um transistor, podendo alcançar uma tensão elevada o suficiente para romper a camada de óxido [10]. As regras da tecnologia utilizada exigem que a proporção entre a área de metal e a área da porta seja menor que um determinado valor limite. Em alguns pontos do layout a área da trilha de metal conectando as portas aos pads foi suficiente para violar essa regra. A solução foi a inclusão de uma via próxima à porta criando uma “ponte” utilizando o nível mais alto de metal disponível – neste caso o metal6 – tal que durante a fabricação a conexão entre a porta e a área grande de metal só ocorre no final do processo e as possíveis cargas acumuladas já foram descarregadas.

IV. SIMULAÇÃO

A simulação do circuito foi realizada em duas etapas. A primeira, considerando somente o esquemático, teve como objetivo mostrar a capacidade do circuito de gerar as formas necessárias. A Fig. 7 mostra a forma trapezoidal genérica e o efeito da variação da tensão ΔV_c no tamanho da mancha de incerteza. A tensão de alimentação utilizada foi de 1.8V e as tensões de polarização – conforme as Fig. 4 e 5 – V_{cn} e V_b iguais a 800mV.

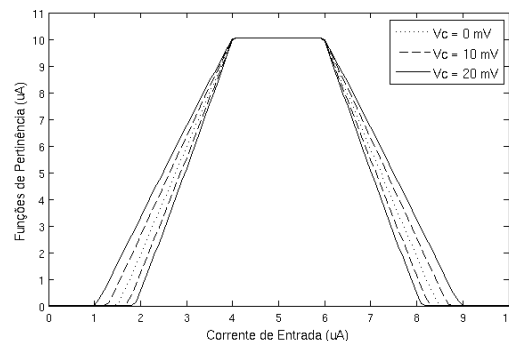


Fig. 7. Funções de pertinência trapezoidais com diferentes manchas de incerteza. Nos 3 casos $I_{T1} = 4\mu$ A, $I_{T2} = 6\mu$ A, $V_1 = 0$ V e $V_2 = 0$ V.

As Fig. 8, 9 e 10 mostram as formas S, triangular e Z, respectivamente, que podem ser obtidas variando as correntes de configuração.

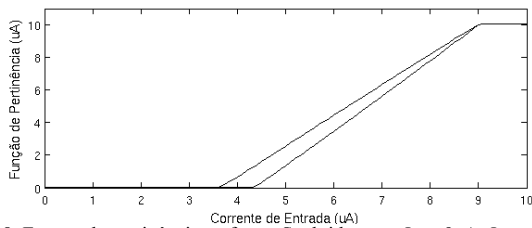


Fig. 8. Função de pertinência na forma S, obtida com $I_{T1} = 9\mu\text{A}$, $I_{T2} = 11\mu\text{A}$, $V_1 = 1.8\text{V}$ e $V_2 = 0\text{V}$.

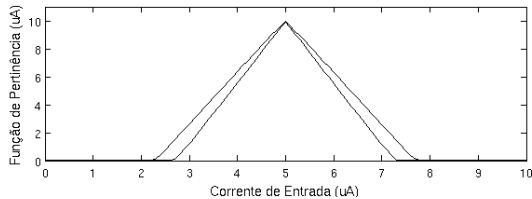


Fig. 9. Função de pertinência na forma triangular, obtida com $I_{T1} = 5\mu\text{A}$, $I_{T2} = 5\mu\text{A}$, $V_1 = 0\text{V}$ e $V_2 = 0\text{V}$.

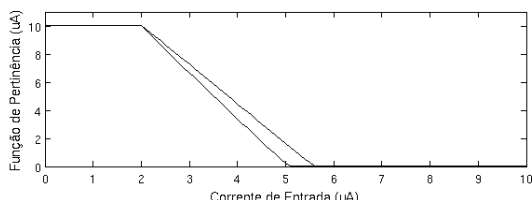


Fig. 10. Função de pertinência na forma Z, obtida com $I_{T1} = 0\mu\text{A}$, $I_{T2} = 2\mu\text{A}$, $V_1 = 0\text{V}$ e $V_2 = 1.8\text{V}$.

Em seguida foi realizada a simulação pós-*layout* – que considera as capacitâncias e resistências parasitas extraídas do *layout*. Utilizando o mesmo arranjo da Fig. 7 somente para $\Delta V_c = 10\text{mV}$, obteve-se o resultado da Fig. 11. É possível observar uma leve diferença entre as funções antes e depois da extração, como se a tensão ΔV_c fosse maior no segundo caso.

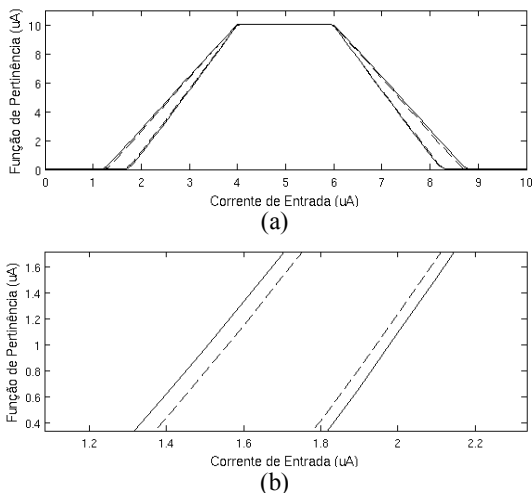


Fig. 11. Função de pertinência na forma trapezoidal, obtida com $I_{T1} = 4\mu\text{A}$, $I_{T2} = 6\mu\text{A}$, $V_1 = 0\text{V}$ e $V_2 = 0\text{V}$. (a) Comparação entre o resultado das simulações antes e depois da extração de parasitas do *layout*. (b) Detalhe da diferença entre as funções: O resultado pré-*layout* está em linha tracejada enquanto o resultado pós-*layout* está em linha contínua.

Como é possível inferir da Fig. 7, a diferença entre as correntes superior e inferior é bastante sensível a ΔV_c , tal que variações da ordem de mV são suficientes para alterar o balanço das correntes. Na construção do *layout*, os pontos conectados a V_{cn} e V_{cp} não são simétricos (diferentes áreas, vias e metais) e as diferentes capacitâncias/resistências parasitas resultantes de tais diferenças de *layout* são suficientes para criar a diferença observada. Mesmo assim, do ponto de vista funcional não há nenhum impedimento, já que uma simples calibração, variando a tensão V_{cp} , permite alcançar o tamanho da mancha de incerteza desejado. Assim, o projeto e a funcionalidade do circuito continuam corretos, mostrando que a arquitetura desenvolvida e implementada é bastante tolerante a eventuais variações de processos de diferentes *foundries*, fornecendo possibilidades de correção por meio de entradas ajustáveis para a geração das diferentes funções de pertinência.

V. CONCLUSÃO

Os controladores utilizando lógica *fuzzy* tipo-2 representam um potencial ainda pouco explorado em ambientes que apresentam incertezas intrínsecas ou incertezas de medida. As poucas implementações analógicas de circuitos fuzzificadores limitam-se a utilizar dois circuitos idênticos para criar as funções superior e inferior ou utilizam um circuito deslocador para criar uma a partir da outra. Neste trabalho foi apresentado um circuito capaz de gerar as duas funções simultaneamente. Foi apresentado também seu *layout* e as considerações sobre *latch-up* e “efeito antena” que tiveram de ser observadas durante sua construção. Por fim, foi confirmada a correta operação do circuito, analisando o resultado das simulações pré e pós-*layout*, avaliando as diferenças decorrentes das capacitâncias e resistências parasitas, bem como a possibilidade de correção, via programação externa, de eventuais discrepâncias de funcionamento do circuito, mantendo-o operacional e plenamente funcional em quaisquer situações.

REFERÊNCIAS

- [1] L. A. Zadeh, “The concept of a linguistic variable and its application to approximate reasoning—I,” *Inf. Sci. (Ny)*, vol. 8, no. 3, pp. 199–249, Jan. 1975.
- [2] J. M. Mendel, H. Hagsras, W.-W. Tan, W. W. Melek, and H. Ying, *Introduction to Type-2 Fuzzy Logic Control*, 1st ed. Hoboken: John Wiley & Sons, Inc., 2014.
- [3] J. M. Mendel, “Type-2 Fuzzy Sets and Systems: An Overview,” *IEEE Comput. Intell. Mag.*, vol. 2, no. 1, pp. 20–29, 2007.
- [4] P. M. S. Rocha Rizol, L. Mesquita, O. Saotome, and G. Botura, “Hardware implementation of type-2 programmable fuzzifier,” in *IEEE 2nd Latin American Symp. on Circuits and Systems (LASCAS)*, 2011, pp. 1–4.
- [5] M. Khosla, R. Sarin, and M. Uddin, “Design of an analog CMOS based interval type-2 fuzzy logic controller chip,” *Int. J. Artif. Intell. Expert Syst.*, vol. 2, no. 4, pp. 167–183, 2011.
- [6] H. Yazdanjoui, H. Feizy, A. Khoei, and K. Hadidi, “Design of a

- Fully Programmable Analog Interval Type-2 Triangular / Trapezoidal Fuzzifier,” in *19th Int. Conf. “Mixed Design of Integrated Circuits and Systems,”* 2012, pp. 243–248.
- [7] A. Mesri, A. Khoei, and K. Hadidi, “Hardware implementation of interval type-2 fuzzy logic controller,” in *21st Iranian Conf. on Electrical Engineering (ICEE)*, 2013, pp. 1–6.
- [8] J. Ramirez-Angulo, S. R. S. Garimella, A. López-Martin, and R. G. Carvajal, “Gain programmable current mirrors based on current steering,” *Electron. Lett.*, vol. 42, no. 10, pp. 559–560, 2006.
- [9] R. Jacob Baker, *CMOS: Circuit Design, Layout, and Simulation*, 3rd ed. Hoboken: John Wiley & Sons, Inc., 2011.
- [10] B. Razavi, *Design of analog CMOS integrated circuits*. New York: McGraw-Hill, 2001.