

Logika cyfrowa

Lista zadań nr 9

Termin: 3 maja 2020

Uwaga! Podczas zajęć należy znać pojęcia zapisane **wytluszczoną czcionką**. W przypadku braku znajomości tych pojęć student może być ukarany punktami ujemnymi.

1. Narysuj schemat obwodu implementującego **deterministyczny automat skończony** (lub **automat Moore’a** z jednobitowym wyjściem) opisany poniższą **tabelą stanów**, ze stanem akceptującym 11:

q	a	q_o
00	0	10
01	0	01
10	0	11
11	0	10
00	1	11
01	1	00
10	1	00
11	1	01

2. Zdefiniuj deterministyczny automat skończony (np. używając tabeli stanów lub **diagramu stanów**) rozpoznający ciągi bitów kończące się sekwencją 1001 lub 1111 (czyli język $(0 + 1)^*(1001 + 1111)$).

Przykładowo, automat powinien rozpoznać ciąg 010111100110011111 oraz jego prefiksy: 01011110011001111, 01011110011001, 0101111001, 0101111.

3. Zaprojektuj obwód, którego wyjście o jest w stanie wysokim wtedy i tylko wtedy, gdy przez cztery kolejne cykle zegara dwa bity wejściowe w_1 i w_2 są równe. Na przykład:

w_1 : 0110111000110

w_2 : 1110101000111

o : 0000100001110

4. Zdefiniuj automat Mealy’ego rozpoznający ciągi bitów, które wśród trzech ostatnich bitów mają nieparzystą liczbę jedynek (czyli język $1 + 01 + 10 + (0 + 1)^*(001 + 010 + 100 + 111)$).

5. Zdefiniuj automat Moore’a rozpoznający, czy liczba jedynek na wejściu jest podzielna przez 4 lub o 1 większa od liczby podzielnej przez 4.

6. Zdefiniuj automat Moore’a rozpoznający ciągi bitów zawierające jednocześnie parzystą liczbę jedynek i parzystą liczbę zer.

7. Narysuj diagram stanów automatu zdefiniowanego następującym kodem w SystemVerilogu:

```
module ckt(  
    input clk, rst, a, b, c,  
    output logic q  
);  
    logic e;  
    always_ff @(posedge clk or posedge rst)  
        if (rst) begin  
            e <= 1'b0;  
            q <= 1'b0;  
        end else begin  
            e <= a || b;  
            q <= e && c;  
        end  
endmodule
```

8. Narysuj diagram stanów automatu zdefiniowanego następującym (zblizonym do tego z poprzedniego zadania) kodem w SystemVerilogu:

```
module ckt(  
    input clk, rst, a, b, c,  
    output logic q  
);  
    logic e;  
    always_ff @(posedge clk or posedge rst)  
        if (rst) e <= 1'b0;  
        else e <= a || b;  
    always_comb q = e && c;  
endmodule
```

9. Pokaż, jak dla dowolnego automatu Moore'a \mathcal{M} zdefiniować równoważny automat Mealy'ego \mathcal{M}' – tzn. taki, że dla każdego $w \in \Sigma^*$ zachodzi $O(\mathcal{M})(w) = O(\mathcal{M}')(w)$.
Podpowiedź: \mathcal{M}' będzie miał taki sam zbiór stanów, jak \mathcal{M} .
10. Pokaż, jak dla dowolnego automatu Mealy'ego \mathcal{M} zdefiniować równoważny automat Moore'a \mathcal{M}' – tzn. taki, że dla każdego $w \in \Sigma^*$ zachodzi $O(\mathcal{M})(w) = O(\mathcal{M}')(w)$.
Podpowiedź: \mathcal{M}' będzie miał więcej stanów, niż \mathcal{M} . Może być pomocny ogólniejszy lemat na temat $O(\mathcal{M})(w, q)$.