

Errare humanum est, perseverare diabolicum!

Nic na świecie nie jest doskonałe. Gdyby udało się wyprodukować rzeczy doskonałe, to wielkość działalności ludzkiej straciłaby sens. Chęć dokonywania ulepszeń, a więc dążenie do doskonałości, jest napędem naszej cywilizacji. W związku z tym zwracam się z uprzejmą prośbą o dostarczanie mi drogą e-mailową wszelkich uwag dotyczących podręcznika: *Podstawy projektowania układów cyfrowych*, a w szczególności wszelkich dostrzeżonych błędów. W miarę nadsyłania informacji o wykrytych błędach zawartość tego dokumentu będzie puchła. Mam nadzieję, że errata objętością nie przewyższy dzieła, którego dotyczy. Jednocześnie sam też krytycznie spoglądam na swe dzieło – co niestety prowadzi do smutnego wniosku, że nie jestem doskonały, ale na pocieszenie mogę dodać, że nie brakuje mi spostrzegawczości.

Autor

Lista dostrzeżonych błędów¹:

str.19 w.10d: Jest: "...możena...", a powinno być "...można..."

str.104–106: Wobec specyfikacji (2.45) na rysunkach: 2.101, 2.102 i 2.103 nie powinno być połączenia między wyjściem O_3 dekodera a bramką realizującą funkcję y_3 (bramką NOR na rys.2.101, bramką OR z zanegowanymi wejściami oraz wyjściem na rys.2.102 oraz bramką AND na rys.2.103). Ponadto wyjście O_5 dekodera powinno być dołączone do bramki realizującej funkcję y_3 .

Łatwiej jest jednak poprawić błąd zmieniając specyfikację (2.45) w następujący sposób: w funkcji y_3 zastąpić 3 przez 5. W konsekwencji w (2.46) trzeba ponownie w y_3 zastąpić 3 przez 5, a w \bar{y}_3 należy zastąpić 5 przez 3.

str.157 w.9g: Jednym z warunków zgodności stanów ze zbioru $\{s^1, s^2, s^3, s^5\}$ jest zgodność pary stanów s^2, s^4 a nie jak napisano: ...oraz s^2, s^3 .

str.229 w.2g: Jest: "Pakunki małe i średnie nie są w stanie zasłonić detektora x_3 ." Oczywiście chodzi o detektor x_2 a nie o x_3 .

str.235, 236 i 237: We wzorze (4.29) w wyrażeniu na q'_1 implikant $x_1\bar{x}_2\bar{x}_3$ powinien zostać zastąpiony implikantem tańszym $x_1\bar{x}_3$. Obecne rozwiązanie nie dość, że jest droższe, to jeszcze prowadzi do powstania hazardu, co łatwo można zauważyć przyglądając się tablicy Karnaugh'a z rys. 4.88 (dokładnie jej kolumnom 110 oraz 100). W konsekwencji należy poprawić nie tylko wzór (4.29), ale również rysunek 4.88 oraz schemat z rysunku 4.91.

str.296 w.13d: Jest: "...przygotowującym...", a powinno być "...przygotowującym..."

¹Przyjęto następujące oznaczenia:

w – wiersz
d – wiersz liczony od dołu strony
g – wiersz liczony od góry strony

Powyższe błędy i niedociągnięcia usunięto w wydaniu książki z 2013 roku. Później wykryłem, że zawartość rozdziału 7 wymaga dodatkowego komentarza, który zamieszczam poniżej.

Na str.313/314 zamieszczono komentarz dotyczący powodu przemieszczenia ładowania (inicjacji) rejestrów $Rejestr_A$, $Rejestr_B$ bądź licznika $Licznik$ przed badanie wartości sygnału $Start$ – modyfikacja dotyczy sieci działań z rys. 7.6, z której w konsekwencji powstała sieć działań z rys. 7.8. Komentarz ten jest jak najbardziej słuszny, ponieważ dotyczy równoległości ładowania rejestrów/licznika i badania sygnału $Start$. Niestety przy realizacji układowej dochodzi do rozsunęcia w czasie tych operacji, więc poprawność prezentowanego rozwiązania trzeba poprzedzić dodatkowym założeniem o rozsunieciu w czasie pojawiania się wartości x_A i x_B oraz sygnału $Start$ (później oznaczanego jako x_s). Prezentowane rozwiązania są prawidłowe jedynie gdy pojawienie się prawidłowych wartości x_A i x_B będzie wcześniejsze o czas t względem pojawienia się x_s . Czas t zależy od tego, czy konstruujemy automat sterujący Moore’a czy Mealy’ego. Niestety na ten czas projektant układu mnożącego nie ma wpływu, to system zewnętrzny musi spełnić wspomniane wymaganie.

Analiza dla układu sterującego Moore’a

Jeżeli układ sterujący Moore’a jest taktowany zegarem jednofazowym, to $t = 0$, a więc żadne dodatkowe założenia nie są potrzebne. Niestety ze względu na to, iż albo sprawdzanie zawartości licznika $Licznik$ (rys. 7.10) albo zawartości rejestru $Rejestr_B$ jest bezpośrednio poprzedzone zmniejszaniem zawartości tego licznika albo przesuwaniem zawartości wspomnianego rejestru, to trzeba zastosować zegar dwufazowy, aby rozdzielić te operacje w czasie. Powodem wprowadzenia zegara dwufazowego jest spowodowanie niejednoczesności wykonywania przez układ operacyjny i układ sterujący ich operacji. Rodzi to jednak problem z załadowaniem argumentów mnożenia na samym początku działania algorytmu. Zegar dwufazowy zastosowany do oddzielnego taktowania układu operacyjnego i układu sterującego Moore’a spowoduje, że modyfikacja sieci działań z rys. 7.6, w wyniku której powstały sieci działań z rys. 7.8, 7.12 i 7.13, a dotycząca przeniesienia mikrooperacji $Rejestr_A \leftarrow x_A$ oraz $Rejestr_B \leftarrow x_B$ lub $Licznik \leftarrow x_B$, spowoduje w pewnych warunkach wadliwe działanie systemu. System będzie działał prawidłowo jedynie gdy układ zewnętrzny będzie wystawiał $x_s = 1$ po czasie t od od wystawienia liczb x_A i x_B , gdzie $t > \frac{1}{2}T_c$ (T_c jest okresem sygnału zegarowego x_c). Przy takim założeniu zostały zaprojektowane układy sterujące Moore’a zarówno te o minimalnej liczbie stanów jak i te wykorzystujące rozdzielacz sterujący oraz układ mikroprogramowany. Jeżeli nie jest spełniona nierówność $t > \frac{1}{2}T_c$, to należy wrócić do sieci działań z rys. 7.6, które spowoduje pojawienie się dodatkowego węzła (stanu) w grafie układu sterującego.

Analiza dla układu sterującego Mealy’ego

Układy sterujące Mealy’ego (minimalny, rozdzielacz i mikroprogramowany) zostały zaprojektowane przy założeniu, że $t > T_c$. Tutaj mamy wprawdzie do czynienia z zegarem jednofazowym, ale należy pamiętać, że zmiana wyjścia automatu Mealy’ego wyprzedza zmianę stanu. Załadowanie rejestrów $Rejestr_A$, $Rejestr_B$ bądź licznika $Licznik$ powodowana aktywnym zboczem zegara, następuje jedynie, jeżeli automat znajduje się w stanie s^a i $x_s = 0$ (rys. 7.39, 7.40). Zmiana x_s na 1 powoduje automatyczne przełączenie wyjścia z $Op0$ na NOp , a więc przygotowanie załadowania rejestrów $Rejestr_A$, $Rejestr_B$ przestanie być aktywne jeszcze gdy układ sterujący będzie w stanie s^a . Tak więc układ sterujący przejdzie do stanu s^d bez załadowania tych rejestrów. Jeżeli układ zewnętrzny nie jest w stanie zapewnić zwłoki $t > T_c$, to można powielić klatkę operacyjną $Op0$ wstawiając kopię przed stan s^d . Oczywiście to rozwiązanie jest równoważne powrotowi do sieci działań z rys. 7.6. Należy podkreślić, że nie powoduje ono wzrostu liczby stanów, węzły grafu automatu sterującego Mealy’ego wry-

sowywane są przed klatkami warunkowymi, a nie operacyjnymi. Wprowadzenie kopii $Op0$ na schemacie rozdzielacza sterującego z rys. 7.64 sprowadzi się do dodania jeszcze jednego wyjścia $Op0$ z tego układu, a konkretnie wyjście to będzie pobierane z wyjścia prawej bramki AND pary bramek AND umieszczonej poniżej przerzutnika a , a realizującej sprawdzenie warunku x_s .

Ponadto wykryto kolejny błąd:

str.313 rys. 7.8: W pierwszej od góry sieci działań klatce operacyjnej jest:

“*Licznik* $\leftarrow x_B$ ”, a powinno być “*Licznik* $\leftarrow x_B$ ”.