Takmer všetky digitálne integrované obvody implementujúce zložitejšiu funkcionalitu než základné logické funkcie majú jeden alebo viac hodinových vstupov. Existuje však aj netypický prístup, ktorý sa snaží hodinám v logických obvodoch vyhýbať.

Všetky zložitejšie integrované obvody vznikajú ako vrstevnatá skladačka - ak opominieme najnižšiu úroveň tranzistorov, obvody sú skladané z hradiel implementujúcich základné logické funkcie, z tých sú postupne skladané stále väčšie a zložitejšie kombinačné obvody. V jednotlivých cestách týchto obvodov vznikajú rôzne veľké oneskorenia, vďaka ktorým sa u zložitejších kombinačných obvodov po zmene vstupných signálov na určitú (prechodnú) dobu ocitajú výstupné signály v navzájom nekonzistetných stavoch (v literatúre najmä pri návrhu FPGA označovaných ako hazardy), čo môže ohroziť funkcionalitu na toto nadväzujúcich blokov. Preto sa častokrát jednotlivé funkčné celky oddeľujú sadou registrov, ktoré zachytia stav predchádzajúceho bloku a prenesú ho na vstup nasledujúceho bloku v okamihu príchodu hodinového signálu. Ten potom musí mať najviac takú frekvenciu, aby sa všetky zmeny v rámci bloku stihli udiať v rámci jednej periódy, berúc do úvahy aj variabilitu oneskorení následkom zmien teploty, napájacieho napätia, ako aj nepravidelnosti samotného hodinového signálu.

Okrem toho sa v digitálnych IO používa aj ďalšia trieda sekvenčných logických obvodov, ktorých úloha je v závislosti na vstupných signáloch prechádzať niekoľkými rozličnými vnútornými stavmi. Tu sa priamo ponúka potreba hodinového signálu, ako zdroj "hnacieho" mechanizmu určujúceho takt týchto zmien.

Toto je dôvod, prečo sa hodinové signály u zložitých digitálnych IO používajú. Prítomnosť pre celý čip jednotného, externe určeného hodinového signálu je jedným zo základných východísk súčasných metodík návrhu IO. Najvypuklejšie je to poznať na návrhu FPGA, kde "synchrónny návrh" je mantrou nesúcou sa od úplného začiatku výuky navrhovania - čo neprekvapuje, ak si uvedomíme, aké nepredpovedateľné a až nečakane dlhé môžu byť návrhovým softwarom vygenerované jednotlivé signálové cesty (a tým aj oneskorenia) v pomerne rigidnej bunkovitej štruktúre FPGA.

Ak by neexistovali záporné stránky takejto synchrónnej logiky, na tomto mieste by sme mohli aj skončiť. Ale existujú.

Najzávažnejším nedostatkom je to, že sa v IO zmení logický stav veľkého množstva hradiel v jednom okamihu, t.j. na hrane hodinového signálu. To spôsobuje špičky v prúdovej spotrebe. Toto má za následok jednak potrebu vykrývať tieto špičky dôsledným filtrovaním (to je ten známy 100nF kondenzátor tesne pri napájacích vývodoch), a tiež si to vyžaduje hrubšie než nevyhnutne potrebné prívody a rozvody napájania po čipe, aby sa zamedzilo poruchám funkcionality kvôli úbytkom napätia. Ďalším neblahým následkom prúdových špičiek je elektromagnetické vyžarovanie. Menším, aj keď v bezpečnostných aplikáciách veľmi závažným následkom je možnosť z merania priebehu prúdovej špičky usúdiť na zmeny vnútorného stavu obvodu (varianta power analysis attack, tzv. switching glitch attack).

Ďalší nedostatok potreby hodinových signálov sa ukázal pri zvyšujúcich sa nárokoch na rýchlosť obvodov, pokračujúcom zväčšovaní integrácie a súvisiacom zmenšovaní šírky čiar, ako aj možnosti vyrábať v absolútnych rozmeroch čoraz väčšie čipy. Veľké množstvo obvodov na ktoré je hodinový signál pripojený a aj kapacita prívodov si vyžaduje mohutné budiče, čo predstavuje viac cennej plochy čipu ako aj zvýšenú spotrebu. Predlžujúce a zužujúce sa vodiče spôsobujú čoraz väčšie oneskorenie, čím hodinový signál v rôznych miestach obvodu môže byť navzájom nesynchrónny. Toto vyžaduje zmenu metodiky návrhu (a súvisiace vysoké náklady do návrhových systémov a nadväznej tvorby základných knižníc obvodov), ako aj vetvenie a pridávanie ďalších budičov - čo znova znamená viac plochy kremíka a väčšiu spotrebu.

Toto všetko vedie k záujmu o návrh asynchrónnych logických obvodov. Nie je to záujem nový - prvé úvahy ba dokonca aj použitie prišli ešte v priekopníckej dobe v 60tych rokoch minulého storočia u počítača ILLIAC II.

Aj keď slovo "asynchrónny" by sa mohlo chápať ako "bez hodinového signálu" nie je to celkom správny pohľad - dôvody pre hodiny uvedené v úvode (potreba resynchronizácie výstupov zo signálových ciest s rôznym oneskorením a sekvenčné obvody) samozrejme pretrvávajú. Čoho sa asynchrónne logika stráni, sú centrálne, externé hodiny. Jednotlivé logické elementy si tu určitým spôsobom "generujú" lokálny hodinový signál samé, podľa vlastných potrieb. Správnejší, aj keď menej používaný, je preto pre takúto logiku výraz self-clocking logic.

Tieto "hodiny" vlastne predstavujú signál potvrdzujúci platnosť dát, ktorý postupuje paralelne spolu s dátami. Jednou z implementácií je oneskorovací obvod, ktorého oneskorenie je rovnaké či o málo dlhšie než oneskorenie najdlhšej signálnej cesty v danom obvode. V mieste, kde sa stretávajú výsledky viacerých takýchto blokov, sa "platnostné" signály združujú pomocou tzv. Mullerovho C-hradla (nazvaného podľa jedného z konštruktérov už spomenutého ILLIAC II), ktorý zmení svoj stav až ked sa zmení stav všetkých jeho vstupov. Problémy sa samozrejme ukrývajú v detailoch, a tým sa práve jednotlivé návrhové prístupy líšia.

Celková metodika návrhu asynchrónnych IO sa teda dosť výrazne odlišuje od konvenčných metodík, a to znamená aj potrebu nových návrhových prostriedkov. To pochopiteľne znamená obrovské množstvo prostriedkov a času, aby sa dospelo do stavu

porovnateľného so stavom návrhu "konvenčných" digitálnych IO. A benefit z tohoto je obvod bez prúdových špičiek, s menším EMI, s menším počtom "horúcich bodov" na čipe, s potenciálne menšou spotrebou, avšak s väčšou plochou (kvôli potrebe "paralelných" hodinových ciest, oneskorovacích prvkov a súvisiacej logiky). Nečudo, že za takýchto okolností sa do tohoto biznisu investori nehrnú, a tak asynchrónna logika ostáva poväčšinou na akademickej pôde, i keď tam zase vznikli aj pozoruhodné kúsky, vrátane zrealizovaných prototypov, napr. tento 16-bitový asynchrónny mikroprocesor, alebo tento asynchrónny klon R2000.

Nájdu sa však aj výminky v komerčnom svete. Jednou z najvýraznejších bola firma Handshake Solutions z technologického inkubátora Philipsu, ktorá pred asi 10 rokmi vypracovala metodiku návrhu, implementovala návrhový systém (zvaný HAST) a vyvinula asynchrónnu verziu populárnej 8051 ako aj Philipsovského rozšírenia 51MX, najmä na interné použitie. Neskôr v spolupráci s ARMom vyvinuli a implementovali aj asynchrónnu verziu ARM9, tá však napriek výrazne zníženej spotrebe nenašla komerčné využitie (ARM sa v tom čase používali dominantne v SoC pre settop boxy, kde nie je tlak na spotrebu, naviac v nich spotrebu častokrát určuje dekódovací hardware a nie procesorové jadro). A tak, po odčlenení NXP, v čase krízy Philips svoj inkubátor spolu aj s Handshake Solutions jednoducho zrušil. Ostal však produkt - spomenuté jadro 51MX - ktoré NXP dodnes používa vo svojich kontaktných aj bezkontaktných smartcard platformách. No a keďže NXP je jedným z dominantných hráčov na tomto poli, následkom nedávnej vlny záujmu o bezkontaktné pasy a identifikačné karty sa zrejme toto jadro stalo komerčne najrozšírenejším asynchrónnym logickým obvodom.