Ipari rendszerek logikai helyességbizonyításának automatizálási lehetőségei

Önálló laboratórium feladat összefoglalója (2. félév)

Tóth Heinemann Zsófia (VCQRJX)
Konzulens: Bartha Tamás
BME Méréstechnika és Információs Rendszerek Tanszék
Internet és Infokommunikációs Alkalmazásai Szakirány, 2008/2009. I. félév

Ipari rendszerek esetében az esetleges meghibásodásoknak igen komoly anyagi, infrastrukturális és esetleg személyi következményei lehetnek. Ezért nagyon fontos mind a *rendelkezésre állás*, mind a *biztonság* megfelelő szintjének elérése, valamint ennek igazolása. Ennek egyik fontos eszköze a megtervezett berendezések ellenőrzése a specifikációra való tekintettel (verifikáció és validáció). Az ipari rendszerek komplexitásánál fogva ezt lehetetlen manuálisan kivitelezni; erre különféle formális (azaz matematikai módszerekkel támogatott) ellenőrzési eljárásokat és leírási módokat találunk a szakirodalomban.

Feladatom olyan eljárások kidolgozása, amivel az előbb leírt ellenőrzési folyamat minél inkább automatizált módon végezhető el. Ennek keretében alkalmazási mintafeladatként a paksi atomerőmű egy újonnan tervezett biztosítóberendezésének megismerését és a működés helyességének alátámasztását végzem el formális eszközökkel. Az önálló laboratórium, majd a diplomatervezés végcélja a megtervezett, valós rendszernek megfelelő modell automatikus származtatása és ellenőrzése; és ennek általánosítása az erőmű (és más ipari alkalmazások) bármely biztosítóberendezésére.

Az erőműben a különböző érzékelők ciklikusan mintavételezik a pillanatnyi állapotot. Az általunk vizsgált berendezés logikája az ezen érzékelőkből jövő jelekre alapozva dönti el, hogy történt-e valamilyen hibaesemény. A kialakítás lényege, hogy szabványos, funkcionális elemként elkészített blokkok működnek közre a jelfeldolgozásban; a kimenet pedig tájékoztat az elvárt működésről vagy a hibaeseményről.

Az általam vizsgált biztosítóberendezésnek kilenc különböző bemenete és két kimenete (egy valódi és egy ellenőrző) van; a logika pedig ötféle modult használ fel, ezek: és-kapu, vagy-kapu, SR flip-flop, pulzusgeneráló modul és egy késleltető-elem.

A téma kezdeti megismerése után azt tűztük ki célul, hogy a fent említett logikát valósítsam meg SAL (Symbolic Analysis Laboratory) modellellenőrző „nyelven”. A SAL egy olyan keretrendszer, melyben különböző eszközökkel (kézi szimuláció, bizonyítandó/ cáfolandó állítások megfogalmazása temporális logikával, stb.) győződhetünk meg egy megírt modell tulajdonságairól. Valamint támogatja a moduláris szemléletet is, ami ez esetben fontos, hiszen a logika többször használ ugyanolyan funkcionális elemet.

Elsőként az SR flip-flop formális modelljét készítettem el és teszteltem működését, majd egy ennek, mint építőelemnek felhasználásával kialakított bináris számlálót. Ezt követte az ún. ONDELAY modul, amely a bemeneti jel adott ideig tartó igaz értéke esetén ad ki egyet a kimeneten. A PULS modul az egyik bemeneti jel igaz értéke esetén ad ki adott ideig tartó „impulzusokat”, a második bemeneti jel felmenő élére viszont nullázza magát. Mindegyik modult szimulációval és LTL logikai kifejezések segítségével teszteltem és tökéletesítettem. A teljes logikát részleteiben raktam össze, hogy könnyebben tesztelhető legyen.

Már az alap építőkockák ellenőrzésénél előtérbe került a klasszikus SAL modulok azon tulajdonsága, miszerint mindig kell nekik valamilyen érték a bemeneteken, és minden ciklusban produkálnak is valami meghatározott kimenetet. Egy kisebb egységnél ez még nem probléma, de a tapasztalat azt mutatta, hogy a teljes logika összeillesztésekor már gondot okoz, ha az egyik modul a másik kimenetéről várná a saját bemenetét, de nincsen semmilyen

köztes, harmadik, „definiálatlan” állapot. Emiatt nem minősíthetem befejezettnek a logika átültetését.

Fentiekből következően a következő feladatom az lesz, hogy megoldást keressek a vázolt problémára. Már több ötlet is felmerült: több állapot bevezetése az bemeneteken és kimeneteken; esetleg más szemlélettel kialakítva a modulokat kimenetét „jelen időben” származtatni (de ez a rekurzív SR flip-flop esetében problémásnak bizonyult eddig). Ha ez a probléma megoldódott, akkor a következhet teljes logika kiterjedt tesztelése.

Hosszú távon számos további lépés merülhet fel. Ki kell választanunk a legalkalmasabb formális leírási módot és analízis környezetet, amiben a logika ellenőrzése hatékonyan elvégezhető, ugyanakkor a modell automatizáltan származtatható. Ki kell dolgozni az ellenőrzési célok automatikus származtatásának módszerét. Végül pedig meg kell oldanunk a talált problémáknak a modellezett rendszerre való „visszavetítését”, ami a rendszerek tervezői számára is hasznos információt szolgáltat.

Mindezen célok nagyon ambiciózusak, így valószínűleg az önálló labor keretében nem jut idő mindegyik megoldására. A munkát ezért diplomaterv keretében tervezem folytatni.